BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

DEVELOPEMENT OF ELECTRONICS AND SOFTWARE ON ROBOTIC VEHICLE CAR4

ROZVINUTÍ ELEKTRONIKY A SOFTWARU NA ROBOTICKÉM VOZIDLE CAR4

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR — **Bc. Mikuláš Michal**
AUTOR PRÁCE

SUPERVISOR — **Ing. Barnabás Dobossy**
VEDOUCÍ PRÁCE

BRNO 2021

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

Assignment Master's Thesis

Institut:	Institute of Solid Mechanics, Mechatronics and Biomechanics
Student:	**Bc. Mikuláš Michal**
Degree programm:	Applied Sciences in Engineering
Branch:	Mechatronics
Supervisor:	**Ing. Barnabás Dobossy**
Academic year:	2020/21

As provided for by the Act No. 111/98 Coll. on higher education institutions and the BUT Study and Examination Regulations, the director of the Institute hereby assigns the following topic of Master's Thesis:

Developement of electronics and software on robotic vehicle Car4

Brief Description:

Project Car4 was founded in 2010 as the end result of bachelor's and master's theses. Since then the vehicle went though further development, as a result of which the software on the main control units has been altered on multiple occasions, remote control unit and also state of charge monitoring unit has been added. The goal of this thesis is to restore and unify the obsolete parts of electronics, which has not been changed since the very beginning of this project. As part of this thesis also the software will be overhauled to ensure smooth steering and independent four–wheel drive.

Master's Thesis goals:

1. Research the field modelling and kinematics of four wheel vehicle. Explore algorithms for no–slip steering.
2. Develop a kinematic model of a vehicle with four driven wheels. Test the created model in a simulation.
3. Evaluate the condition of current hardware elements and replace obsolete electronics components. The evaluation should concern the following hardware units:
– power electronics
– signal electronics
– wheel’s speed of rotation measurement
4. Implement and test the new hardware elements on the Car4 vehicle.
5. Develop a control algorithm that ensures independent adjustment of the speed of rotation of the wheels, the new control algorithm should also ensure no–slip steering.

Recommended bibliography:

VALÁŠEK, M.: Mechatronika, Vydavatelství ČVUT 1995.

VEJLUPEK, J.: Vývoj elektroniky pro řízení trakce experimentálního vozidla. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2010.

VADLEJCH, F.: Konstrukce podvozku experimentálního vozidla se čtyřmi řízenými koly. Vysoké učení technické v Brně. Fakulta strojního inženýrství, 2010.

Deadline for submission Master's Thesis is given by the Schedule of the Academic year 2020/21

In Brno,

L. S.

prof. Ing. Jindřich Petruška, CSc.
Director of the Institute

doc. Ing. Jaroslav Katolický, Ph.D.
FME dean

Abstrakt

Tato diplomová práce se zabývá revizí elektroniky robotického vozidla car4 se zaměřením na měření rychlosti otáčení kol, výkonovou a řídicí elektroniku. Car4 posloužilo jako základ pro již přes 20 diplomových prací od roku 2010. To znamenalo, že některé základní aspekty car4 byly zastaralé. Bylo vytvořeno schéma elektroniky vozidla, které bylo dále využito pro vývoj a výrobu nového hardwaru. Komponenty byly poté otestovány a implementovány na vozidlo. Dále byl vytvořen kinematický model 4WS vozidla za použití Ackermanovy geometrie, který byl implenetován a otestován na car4. Dále by měl sloužit jako základ řídicího algoritmu pro budoucí vývoj car4.

Summary

In this thesis a revision of electronics with focus on wheels' speed of rotation measurement, power and control electronics of robotic vehicle car4 was performed. Car4 has been used as a base for over 20 theses since 2010. This meant that some of the core elements of car4 became decrepit. A schematic of on board electronics was created and used to develop and manufacture new hardware. The new components were then tested and implemented onto the vehicle. A kinematic model of 4WS vehicle using Ackerman geometry was also created and implemented and tested on car4 to serve as base of control algorithm for future development of car4.

Klíčová slova

Návrh DPS, 4WS kinematika, Akermanova geometrie, PID regulátor, MCU, H-můstek, Magnetický enkodér

Keywords

PCB design, 4WS kinematics, Ackerman geometry, PID controller, MCU, H-bridge, Magnetic encoder

Bibliographic citation

MICHAL, M. *Developement of electronics and software on robotic vehicle Car4*. Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2021. 63 pages, Master's thesis supervisor: Ing. Barnabás Dobossy.

Rozšířený abstrakt

Úvod

Car4 je experimentální elektrické vozidlo sestavené v mechatronické laboratiři (= Mechlab) Fakulty strojního inženýrství Vysokého učení technického v Brně. Za vnikem tohoto vozidla stálo v roce 2010 pět závěrečných prací studentů oboru mechatroniky. Vzniklé vozidlo mělo pohon na všechny čtyři kola (4WD) pomocí čtyř stejnosměrných motorů (= DC motorů). Dále vozidlo disponovalo nezávislým natáčením všech čtyř kol (4WS) pomocí čtyř servomotorů. V průběhu let sloužilo car4 jako základ pro mnoho dalších závěrečných prací, které se mimo jiné zabívaly monitorováním stavu baterie, implementací SLAM algoritmů (= Simultaneous localization and mapping) a plánování trajektorie 4WD/4WS vozidla.

Od vzniku car4 uplynulo již 11 let a během vývoje vozidla bylo na car4 provedeno četné množství změn. Bylo rozhodnuto, že je potřebná revize stávající elektroniky vozidla se zaměřením na výkonovou elektroniku, řídicí elektroniku a měření rychlosti otáček kol. Na tuto revizi se zaměřuje tato diplomová práce. Výsledkem této revize by měla být obnovéná a unifikovaná základní elektronika. Dalším výstupem této diplomové práce má být řídicí algoritmus, který zajistí bezprokluzové zatáčení a nezávislé ovládání všech čtyř kol.

Postup řešení

V první části této práce byla provedena rešeršní studie kinematických modelů vozidel. Tyto modely bylo rozděleny na holonomní a neholonomní podle výskytu neholonomních vazeb, přičemž byl vysvětlen rozdíl mezi těmito druhy vazeb. Bylo popsáno rozdílné využití těchto druhů modelů a dále byly popsány příklady kinematických modelů vozidel z obou skupin.

Byl analyzován stav elektroniky na vozidle car4. Hlavní řídicí jednotka (= MCU) byla původní z roku 2010 a byla schopná komunikovat s ostatními jednotkami na vozidle, které byly: modul dálkového ovládání, magnetické enkodéry, jednotky pro ovládání DC motorů a jednotky pro připojení servomotorů. MCU bylo schopné ovládat jen dva stejnosměrné motory a dva servomotory. Dvě jednotky pro ovládání DC motorů s proudovými čidli byly přítomny na vozidle. Obě byly také z původního návrhu a každá byla schopna ovládat jeden DC motor. Dvě jednotky pro připojení servomotorů byly přítomny na vozidle. Každá byla schopná připojit dva servomotory. Opět obě jednotky byly z původního návrhu z roku 2010. Pro měření rychlosti kol byl na každém DC motoru umístěn magnetický enkodér, který snímal natočení rotoru pomocí přilepeného magnetu. Tento způsob měření otáček motoru byl 2 roky starý. Dále byly na vozidle dvě jednotky, které snižovaly napětí baterie car4. Jedna jednotka byla dedikována pro napájení servomotorů. Druhá jednotka snižovala napětí pro řídicí elektroniku na vozidle. Stav elektroniky na vozidle byl vizualizován v blokovém diagramu. Dále byla rozvedena vize pro budoucí vývoj vozidla car4, podle které byly formulovány potřebné změny.

V další kapitole byly odvozeny a testovány dva kinematické modely. První model popisoval kinematický řetězec pro ovládání natočení kol. Tento model byl vytvořen, protože dříve vytvořený model nedostatečně reprezentoval mechanismus natočení kol. Nový model (stejně jako starý) popisuje mechanismus natočení kol jako čtyřkloubový mechanismus. Z vytvořeného modelu byl stanoven pracovní interval, který byl porovnán s pracovním intervalem starého modelu. Byly zjištěny značné rozdíly při větším natočení kola. Dále byl vytvořen holonomní model vozidla car4, který sloužil k vypočítání potřebných natočení a rychlostí jednotlivých kol pro dosažení bezprokluzového řízení. Tento model využil Ackermanovy geometrie.

V následující kapitole byl popsán proces návrhu nové elektroniky. V první řadě byl vytvořen nový blokový diagram navrhované elektroniky. Byly vytvořeny seznam požadavků na novou řídicí jednotku, čtyři nové H-můstky a jednu novou jednotku pro připojení servomotorů. Dále byly popsány procesy návrhů nové elektroniky pro řídicí jednotku a jednotku pro připojení servomotorů. H-můstky nebyly nakonec vyrobeny kvůli jejich složitosti a časovému omezení. To znamenalo, že implementovaná elektronika na vozidle car4 se lišila od návrhu. Pro vizualizaci byl vytvořen blokový diagram, který popisuje aktuální stav elektroniky na vozidle. Hlavní změnou byla přítomnost pouze dvou jednotek pro řízení DC motorů.

V poslední kapitole byl popsán vývoj a implementace nového řídicího algoritmu. Dále jsou v této kapitole popsány testy, které byly provedeny pro ověření správné funkčnosti jednotlivých části řídicího algorimu. Jakožto základ posloužil již dříve vytvořený model v prostředí simulink, který sloužil jako řídicí algoritmus pro car4. Z tohoto modelu byl posléze vygenerován kód v jazyce C, který se nahrál do řídicí jednotky na car4. Tato kapitola byla rozdělena na tři části, které se zaměřují na:

- Regulace otáček DC motorů
- Ovládání natočení kol
- Implementace Ackermanovy geometrie

V případě regulace otáček DC motoru bylo zjištěno, že při velkých požadovaných změnách otáček se u přechových dějů vyskytují poklesy otáček. U DC motorů jsou otáčky motoru přímo úměrné na napětí na motoru. Při připojení motorů na laboratorní zdroj napětí, se motory chovaly dle očekávání. Bylo tedy usouzeno, že za poklesy otáček může baterie car4, konkrétně její stáří a opotřebení. Pro regulaci otáček byl zvolen PI regulátor, pro který byly parametry P a I stanoveny experimentálně.

Během testování nového modelu pro ovládání natočení kol bylo zjištěno, že oba modely mechanismu natočení kol (starý i nový) jsou neoptimální, neboť natočení kol neodpovídala ani jednomu z modelů. Po podrobné analýze mechanimu natáčení kol bylo zjištěno, že osa rotace natočení kola není v rovině s hřídelí servomotoru. Tudíž tento mechanismus nejde popsat ve 2D prostoru. Pro ovládání natočení kol bylo provedeno měření. Ze získaných hodnot byla vytvořena 1D vyhledávací tabulka, která sloužila pro ovládání natočení kol.

Nakonec byl implementován kinematický model využívající Ackermanovu geometrii. Samotný model má tři vstupy a osm výstupů. Algoritmus samotný byl příliš výpočetně náročný, aby se mohl implementovat přímo. V našem případě bylo rozhodnuto, že se car4 bude řídit jako 2WD/2WS vozidlo. Z tohoto rozhodnutí vyplynula zjednodušení, která společně s vypozorovanými vlastnostmi modelu, umožnila implementovat tento model

za použití dvou 1D vyhledávacích tabulek. Tímto způsobem se značně snížila výpočetní náročnost tohoto algoritmu.

Závěr

V této diplomavé práci byl vytvořen nový řídicí algoritmus pro experimentální elektrické vozidlo pojmenované car4. Tento algoritmus by v budoucnu měl sloužit jako základ pro budoucí vývoj vozidla car4 a pro implementaci vyšších řídicích algoritmů. Nový řídicí algoritmus je založen na Ackermanově geometrii a obsahovuje regulaci rychlosti kol pomocí PI regulátorů. Tento algoritmus zajišťuje jízdu bez prokluzu.

Byla také provedena revize elektroniky car4. Na základě provedené revize byl vytvořen návrh nové elektroniky. A podle stanoveného návrhu byla vytvořena nová řídicí jednotka a jednotka pro připojení servomotorů. Tyto jednotky byly posléze otestovány a implementovány na car4. V návrhu nové elektroniky byly také čtyři jednotky pro ovládání DC motorů. Vzhledem k velké časové náročnosti této práce nebyly tyto jednotky vytvořeny.s

I hereby declare that this master's thesis is my own original work. Wherever work of others was used specific reference states this fact. Additionally I state that this thesis wasn't submitted anywhere else for consideration for any other qualification.

Mikuláš Michal

Brno . . . .

I would like to thank my advisor Ing. Barnabás Dobossy for his guidance and patience. I would also like to thank my friends, family and my partner for supporting me not only during the work on this thesis.

Mikuláš Michal

Contents

1 Introduction

Ever since the dawn of men people thought about different ways on how to transport themselves or materials around in order to make their lives easier. Development in many aspects of transportation was made. For example from small wooden boats capable of transporting few people to giant metal tankers capable of transporting hundreds of tones of cargo [22]. From the early biplanes up to spacecrafts that transport people outside of our planet. The invention of steam engine that forever changed the human society. Humankind showed that it is both determined and capable of great feats. Nowadays more and more effort is directed to electric vehicles over standard internal combustion engine vehicles. Electric vehicles offer more efficient way of transforming energy into kinematic energy that the standard combustion engine. Although there are still many issues that need to be tackled in order for the electric vehicle to be deployed in a larger scale. Things like decreasing the charging time and adding more charging stations.

Car4 is an experimental electrical vehicle with four wheel drive (4WD) and with four wheel steering (4WS). The project car4 started in 2010 under mechatronics laboratory (mechlab) at FME BUT. It's purpose is to test and develop systems used in automotive industry. Since then many bachelor's and master's theses were done on car4. For example remote control has been added, electronics to monitor and evaluate the battery status have been implemented and testing of SLAM algorithms was performed.

It has been 11 years since the start of the project and a revision of car4's electronics was needed. This thesis focuses on this revision so that the core electronics are both updated and unified. This thesis also focuses on software changes so that car4 can be driven with no-slip steering.

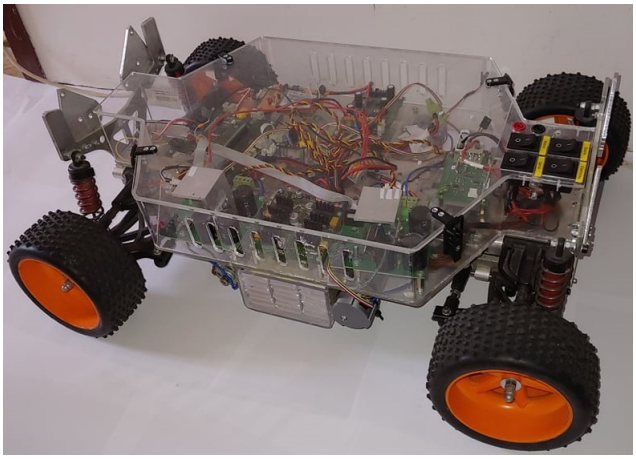


Figure 1.1: Experimental electrical vehicle car4

2 Theoretical survey

In this chapter the results of a theoretical survey concerning different types of kinematic models used to describe four wheeled vehicles was described.

It is important to note that in order to use kinematic models to compute positions, velocities and accelerations several assumptions must be kept in mind:

- Transient events are neglected
- It is presumed that wheels do not lose traction
- Motors must posses more than enough required power

2.1 Types of kinematic models

Kinematic models can be divided into two groups of holonomic and nonholonomic models [7]. The terms holonomic or nonholonimic describe constraints. Nonholonomic constraints are defined for the velocities but cannot be formulated for the positions. In order to determine that a constraint is either holonomic or nonholonomic several approaches can be taken:

- Integration of equations defining restrictions of movement of system
- Reduction of degrees of freedom (=DOF)
  - Holonomic constraints reduce the DOF of a system by whole values of DOF
  - Nonholonomic constraints reduce the DOF of a system by halves of DOF
- In nonholonomic system the order of different movements have impact on the final position of the system

The easiest way to show the difference between the two types of constraints is the last above mentioned method. A simple holonomic system is for example a planar RR manipulator. In fig. 2.1 such planar RR manipulator and the behaviour of this system when two movements have been performed in different succession is shown. The movements for the top three schematics are that the link number 1 sets it's angle α_1 from 0 to $alpha_{1F}$ and then the link number 2 sets it's angle $alpha_2$ from 0 to $alpha_{2F}$. In case of the bottom three schematics the movements are the same except the link number 2 moves first.

An example of nonholonomic system is for example 2WD and 2WS vehicle (standard car). A model of such a car and the different behaviour that can be observed when two movements have been performed in different order can be seen in fig. 2.2. The first movement performed by the modeled vehicle is that it's wheels rotate causing the vehicle to move and the second movement is that the wheels are steered. This order can be seen in the top three schematics. The bottom three schematics depict the behaviour of the system when the movements are performed in reversed order.

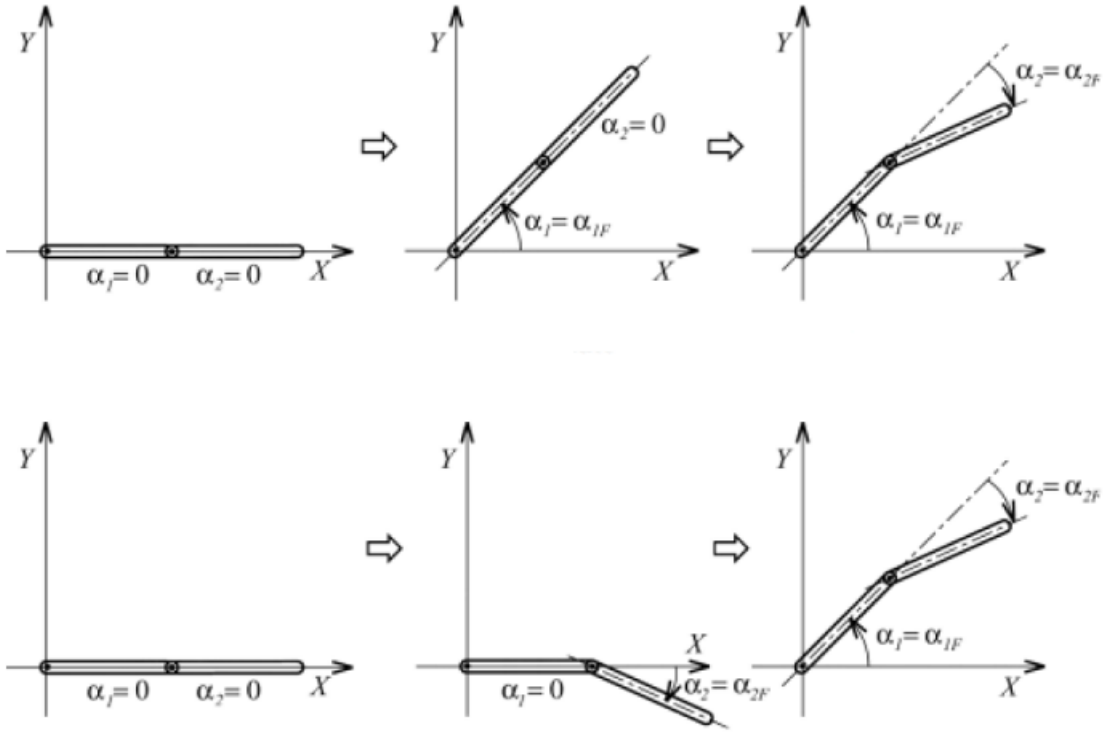


Figure 2.1: Example of holonomic system - planar manipulator [11]

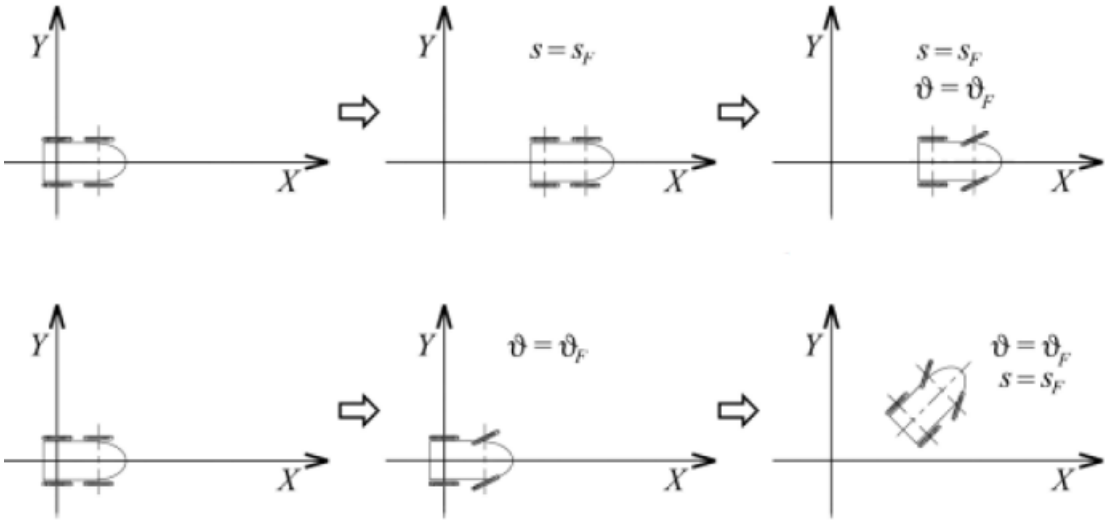


Figure 2.2: Example of nonholonomic system - car [11]

2.2 Models of moving vehicle

These models contain nonholonomic constraints and thus the kinematic models themselves are also called nonholonomic. The vehicle is reduced in some way into often planar space although 3D models also exist. These models perform the task of trajectory planning.

2.2.1 Bicycle model

An example of a kinematic model of a wheeled vehicle could be a that of a standard bicycle [10]. The front wheel is used for steering and the rear wheel is driven. A car with a driven rear axle (2WD) and front steering (2WS) could be modeled in this way as both the rear and front axles were reduced into single wheels. This way four wheeled vehicle

can be represented by a simplified one-track model. The model can be seen in fig. 2.3. The stability of ride is not addressed by the model. In the schematic the point C is the ICR. Local coordinate systems of the front and rear wheel are also displayed. The state of the bicycle is defined by the position and rotation of the rear wheel (point S) x, y, φ, angle of the steering wheels ψ and by the angular position of the rear wheel ϑ.

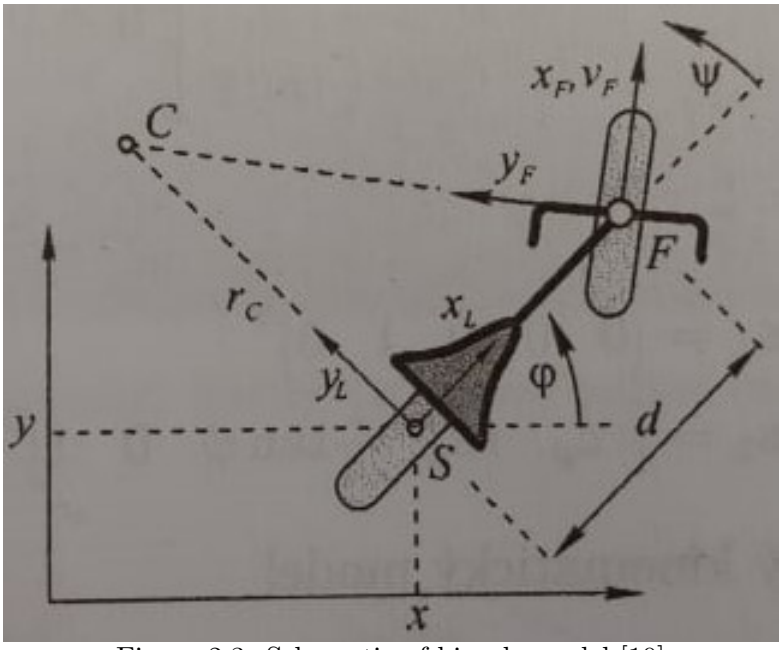


Figure 2.3: Schematic of bicycle model [10]

$$\mathbf{q} = [x \;\; y \;\; \varphi \;\; \psi \;\; \vartheta]^T \tag{2.1}$$

The radius of the rear wheel to the ICR r_c and the relationship of front and rear velocities v_F, v_S are described in eqs. 2.2 - 2.5. Where d is the distance between the wheels.

$$r_c = d \tan(90° - \psi) \tag{2.2}$$
$$v_S = \dot{\varphi} r_c \tag{2.3}$$
$$v_F = \dot{\varphi} d \frac{1}{\sin \psi} \tag{2.4}$$
$$v_S = v_F \cos \psi \tag{2.5}$$

The nonholonomic constraints are formulated in eqs. 2.6 - 2.8 and the same constraints but in global coordinate system in eqs. 2.9 - 2.11.

$$\dot{y_F} = 0 \tag{2.6}$$
$$\dot{y_S} = 0 \tag{2.7}$$
$$\dot{x_L} = \dot{\vartheta} r_c \tag{2.8}$$

$$\dot{x} - r_c \cos\varphi \dot{\vartheta} = 0 \tag{2.9}$$

$$\dot{y} - r_c \sin\varphi \dot{\vartheta} = 0 \tag{2.10}$$

$$\dot{\varphi} - \frac{r_c}{d} \tan\psi \dot{\vartheta} = 0 \tag{2.11}$$

The equations of the model can then be written in matrix form (eq. 2.12). Where the input $u_1 = \dot{\psi}$ is the speed of steering and the input $u_2 = \dot{\vartheta}$ is the speed of the driven wheel.

$$\dot{\mathbf{q}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} r_c \cos\varphi \\ r_c \sin\varphi \\ \frac{r_c}{d} \tan\psi \\ 0 \\ 1 \end{bmatrix} u_2 \tag{2.12}$$

2.2.2 Pseudobicycle model

Another way how to model a vehicle is using a so called pseudobicycle [3]. The schematic of the pseudobicycle can be seen in fig. 2.4. Again both the front and the rear axle of a four wheeled vehicle are reduced into single wheels. In this model both wheels can be steered and the front wheel is driven (simplification for 4WS and front 2WD vehicle).

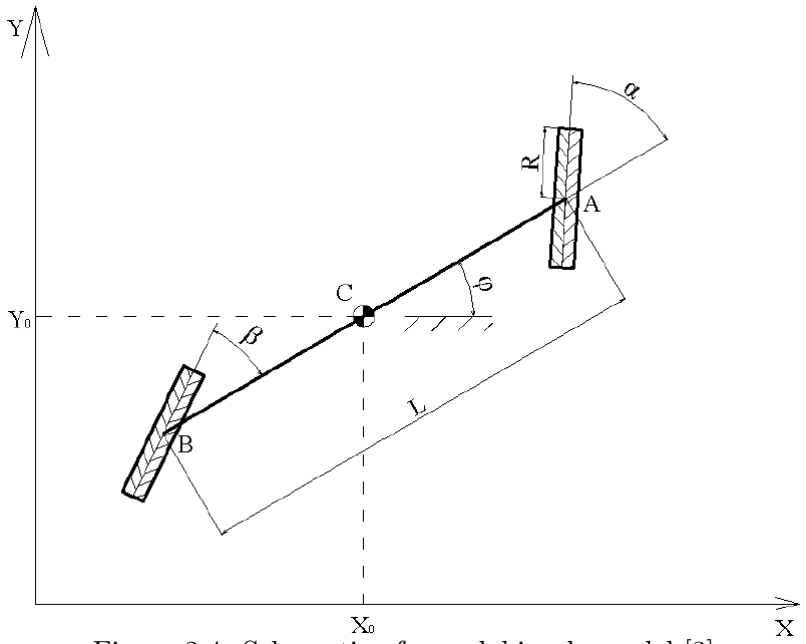


Figure 2.4: Schematic of pseudobicycle model [3]

This model is describe by the state variables of the position of the pseudobicyle (point C) x, y, it's orientation φ, the steering angle of the front wheel α, steering angle of the rear wheel β and the angular position of the driven front wheel ϑ.

$$\mathbf{q} = [x \;\; y \;\; \varphi \;\; \vartheta \;\; \alpha \;\; \beta]^T \tag{2.13}$$

Similarity between the bicycle model is apparent. The main difference is of course that both wheels can be steered meaning that the computation of the ICR position is much more complex as it is a function of both wheels' steering angles α and β. The derived equation of the model is 2.14 [3]. Where R is the radius of the front wheel and L is the length of the pseudobicycle. The input $u_1 = \dot{\alpha}$ is the speed of the steering of the front wheel, $u_2 = \dot{\beta}$ is the speed of the steering of the rear wheel and $u_3 = \dot{\vartheta}$ is the speed of the driven front wheel.

$$\dot{\mathbf{q}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 + \begin{bmatrix} s_{31} \\ s_{32} \\ s_{33} \\ 1 \\ 0 \\ 0 \end{bmatrix} u_3 \tag{2.14}$$

$$s_{31} = -R\frac{\sin\varphi\cos\beta\sin\alpha + \sin\varphi\sin\beta\cos\alpha - 2\cos\beta\cos\varphi\cos\alpha}{2\cos\beta} \tag{2.15}$$

$$s_{32} = R\frac{2\cos\alpha\cos\beta\sin\varphi + \cos\varphi\cos\beta\sin\alpha + \cos\varphi\sin\beta\cos\alpha}{2\cos\beta} \tag{2.16}$$

$$s_{33} = R\frac{\cos\beta\sin\alpha - \sin\beta\cos\alpha}{2L\cos\beta} \tag{2.17}$$

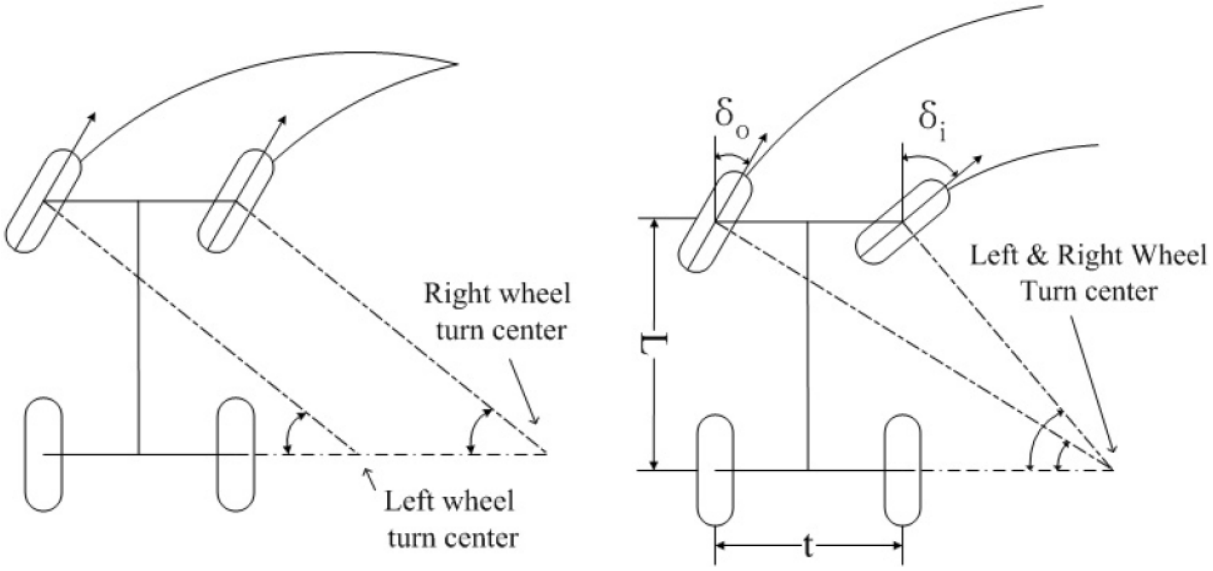
2.3 Models of stationary vehicle

These models are used to compute the required wheels' rotations and the required wheels' angular velocities in order to achieve no-slip control. The models themselves are not in motion meaning that parts of the model do not change coordinates based on input. This means that all of the constraints can be interpreted as holonomic and thus the whole model can be called holonomic.

In case of two-track vehicles (such as a typical car) the kinematics of axles need to addressed. The reason is that when the vehicle turns it is beneficial when it does so around a single point in space called instant center of rotation (ICR). For each wheel the ICR can lie anywhere on it's axis of rotation. So in case of the pseudobicycle there is always one defined ICR. But in case of a 4WS vehicle when for example parallel steering is used (fig. 2.5a) then except for forward motion there isn't a single defined ICR by the rotation of the wheels which causes slip. Similar problem has to be addressed with the speed of individual wheel.

2.3.1 2WS Ackerman geometry

This problem has been addressed by Rudolf Ackerman in the 19th century. Ackerman steering principle (commonly referred to as Ackerman geometry) of 2WS (fig. 2.5b) defines the rotations of the front inside tire and the front outside tire in a turn so that no-slip turning can be achieved (eqs. 2.18 - 2.19, [8]). Since only the front wheels are rotated the ICR will always lie on the axis of rotation of the rear wheels.



(a) Parallel arm steering (b) 2WS Ackerman steering principle

Figure 2.5: Comparison of 2WS steering [8]

$$\delta_0 \cong \frac{L}{R + \frac{t}{2}} \tag{2.18}$$

$$\delta_i \cong \frac{L}{R - \frac{t}{2}} \tag{2.19}$$

2.3.2 4WS Ackerman geometry

A more generalized formula of Ackerman geometry can be defined which can then be used to control a 4WS vehicle [8]. In this case the ICR is no longer bound to the axis of rotation of forward facing rear wheels and can be placed in different positions in the planar space. This way smaller turn radius can be achieved than with a 2WS vehicle when the rear wheels are rotated in contrast to the front wheels resulting in a more low speed maneuverability.

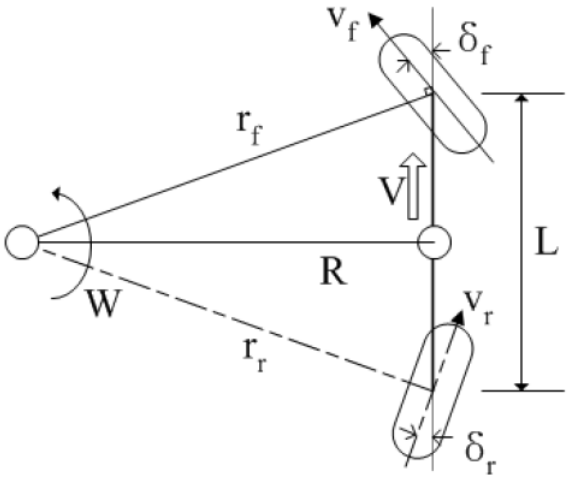


Figure 2.6: Stationary pseudobicycle model [8]

The proposed way in [8] is that the vehicle is approximated using a stationary pseudobicycle model (fig. 2.6). It is presumed that the ICR lies on a line perpendicular to the center of the pseudobicycle's rim. The eqs. 2.20 - 2.21 describe the proportional relation between the steering angle of the front wheel and the steering angle of the rear wheel. The turn radius can then be computed using eq. 2.22. Where L is the length of the vehicle and R is the is the distance of the ICR and the rim of the pseudobicycle.

$$\delta_r = \xi \delta_f \tag{2.20}$$

$$\delta_f + \delta_r = \delta_f + \xi \delta_f = \delta_f(1 + \xi) = \frac{L}{R} \tag{2.21}$$

$$R = \frac{L}{\delta_f(1 + \xi)} \tag{2.22}$$

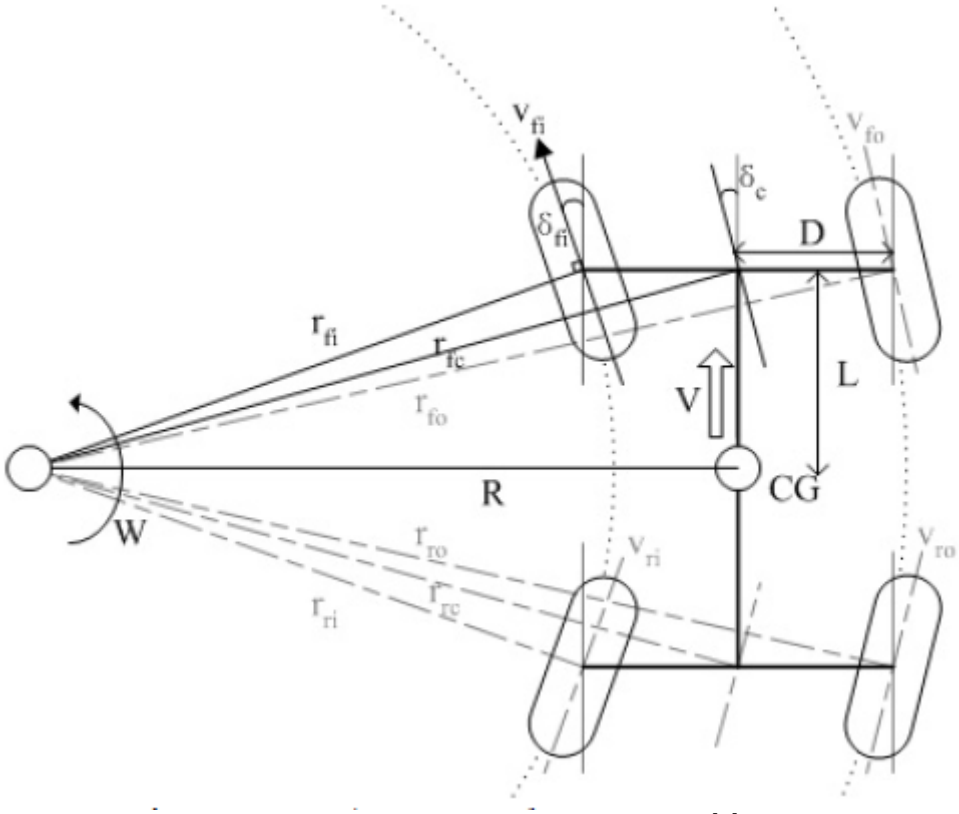


Figure 2.7: 4WS Ackerman geometry [8]

Using the pseudobicycle model with input of the front steering angle δ_f the turning radius of the vehicle can be computed using eq. 2.23 and the angular velocity of the vehicle using eq. 2.24.

$$R = \frac{L}{\tan \delta_c} \tag{2.23}$$

$$\omega = \frac{V}{R} \tag{2.24}$$

The distances between the inner front wheel and the ICR and the outer front wheel and the ICR could then be computed using eqs. 2.25 and 2.26. Using these distances the required speeds of both inner and outer front wheels can then be calculated using eqs. 2.27 and 2.28.

$$r_{fi} = \sqrt{(R - D)^2 + L^2} \tag{2.25}$$

$$r_{fo} = \sqrt{(R + D)^2 + L^2} \tag{2.26}$$

$$v_{fi} = r_{fi}\omega \tag{2.27}$$

$$v_{fo} = r_{fo}\omega \tag{2.28}$$

The needed steering angles of the front wheels can be computed using trigonometric *arcus tangens* (eqs. 2.29 and 2.30).

$$\delta_{fi} = \arctan \frac{L}{R - D} \tag{2.29}$$

$$\delta_{fo} = \arctan \frac{L}{R + D} \tag{2.30}$$

Finally the the require steering angles and speeds of the rear wheels can be computed using eqs. 2.31 - 2.34 since the ICR lies on an axis of symmetry of the model.

$$v_{ri} = v_{fi} \tag{2.31}$$

$$v_{ro} = v_{fo} \tag{2.32}$$

$$\delta_{ri} = \delta_{fi} \tag{2.33}$$

$$\delta_{ro} = \delta_{fo} \tag{2.34}$$

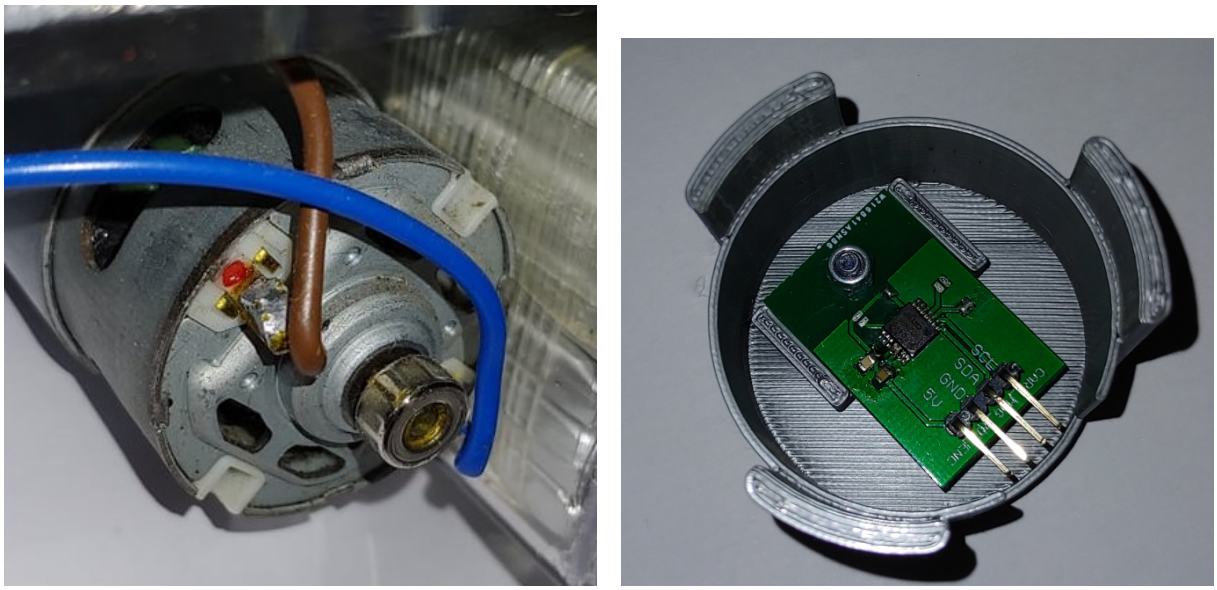
3 Analysis of car4 electronics

In this chapter the state of the car4 core electronics at the beginning of this thesis were analyzed. These electronics can be divided into three categories. Speed measuring sensors which send data into the MCU that processes the data, handles communications and generates outputs to HW control units.

3.1 Speed of rotation measurement

In order to measure the speed of each wheel magnetic encoders *AS5048B* were used. *AS5048B* is an integrated circuit using four hall sensors to detect the angular position of a diametrically polarized magnet. *AS5048B* utilize I2C communications to send the measured data with 14bit resolution. This means that this magnetic encoder is able to measure 16,384 different positions. The device is capable of running on both $3.3\,V$ and $5\,V$.

In our case the magnets were glued to the DC motors' shafts (fig. 3.1a). The *AS5048B* itself was soldered onto a PCB. This PCB was manufactured as the manufacturer's datasheet [13] recommended for $5\,V$ power source. This PCB was held inside a 3D printed case using a single screw as can be seen on fig. 3.1b. This case was then mounted upon the motor with the glued on magnet.



(a) Magnet on rotor (b) Magnetic encoder in case

Figure 3.1: Speed measuring setup

Communication of all four encoders was then connected to a single unit. This unit contained required pull-up resistors. The power of I2C communication was connected to

a power unit providing the required $5\,V$ and the remaining SDA, SCL and Ground lines were connected to the MCU.

The implemented way of measuring the speed of rotation was employed in the year 2019 so it was only 2 years old at the time of writing this thesis. The employed *AS5048B* with it's 14 bit resolution was more than capable of satisfy the requirements of car4. The state of the PCBs and connections was in very good shape overall. The only inconvenience was the fact that the power for the I2C communication was brought from a power unit whilst the remaining four wire were connected to the MCU which added complexity to the cable management. This could be addressed with the new possible MCU design by adding a connector with all four routes. In the end it has been decided that this part of car4's electronics doesn't need to be replaced.

3.2 Main control unit

At the beginning of work on this thesis car4 had one control unit made by the initial creators of car4. This MCU was named *Axle Control Unit 44* - ACU44 and can be seen in fig. 3.2. In the original design [6] there were 2 of these control units as a single one is capable of controlling only 2 DC motors and 2 servomotrs. ACU44 was composed of two double-layer PCBs that can be plugged-in to each other. The *first PCB* contained:

- dsPIC33FJ128MC803 microcontroller
- $20\,MHz$ crystal
- Programming and debugging port
- Reset button
- LF33CV - $3.3\,V$ voltage regulator

The *second PCB* contained the following components:

- L7805CV - $5\,V$ voltage regulator
- MAX3232 - RS232 transceiver
- CAN bus transceiver
- 2x connectors each for DC motor and servomotor control
- 2x QEI connectors
- SPI connector
- I2C connector

The PCBs themselves were in a decent shape. A big problem for future use was the limited number of DC motors and servomotors that this unit can control. This limitation prevented the utilization of the full potential of 4WS and 4WD that car4 possesses. This meant that the design and creation of a new MCU was required.

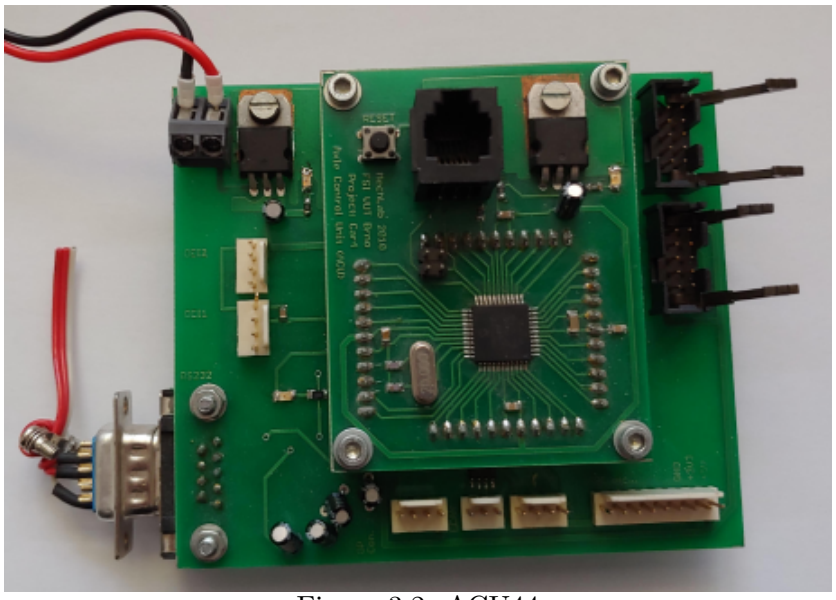


Figure 3.2: ACU44

3.3 Hardware control units

3.3.1 Servomotors interface unit

This unit received PWM signal required by the servomotors for angular position control. The power was received by a second power unit on car4 which was dedicated to servomotors. This power unit was powered by the present 22.2 V battery and droped it to 6 V which were required by the servomotors. This PCB can be seen in fig. 3.3.

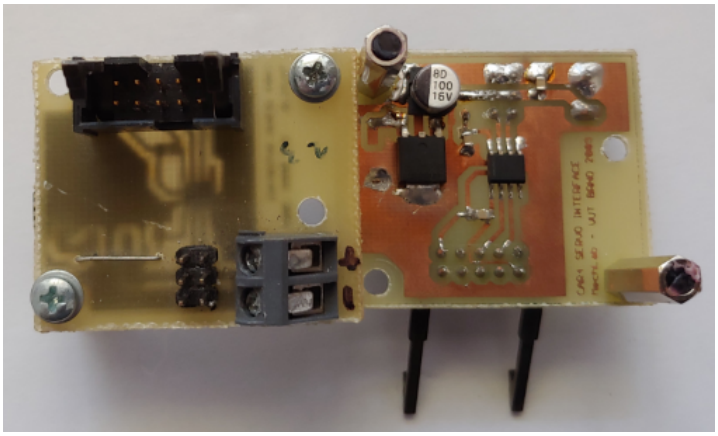


Figure 3.3: Bottom and top view of the servomotor interface unit

One unit was capable of controlling two servomotors. Two units were present on car4 at the time of this revision. The unit contained:

- ISO7220C - digital isolator
- 7805CG - 5 V voltage regulator for the digital isolator
- Power connector
- Connector for PWM signal from the MCU

The two present units functioned without problem. But since they were identical this meant that both of them used the same positions of connector to input PWM signals that control the servomotors. This meant that in order to connect both of them to a single MCU alteration of the cables would be needed. It has been decided that these two servomotor interface units would be replaced by a single one. This way space could be saved and the new servo interface unit could be connected to the new MCU without additional alternation of the connector which in turn reduces the complexity of cable management.

3.3.2 ISL Power boards

This PCB was an H-bridge which was meant to control a DC motor. The unit was powered directly by the battery. The inputs to this unit were PWM, DIR and DIS to control the motor and it outputted measured current flowing through the H-bridge using a LEM current sensor which was powered by $5\,V$ input voltage from the MCU.

The list of components on this PCB is as follows:

- ISL83204A - full bridge FET driver
- LEM-LTS 25NP - current gauge
- ISO7241 - digital isolator
- L7812CV - $12\,V$ voltage regulator for the ISL83204A
- ST 78L05 - $5\,V$ voltage regulator for the ISO7241
- 4x IRFB4115 power MOSFETs

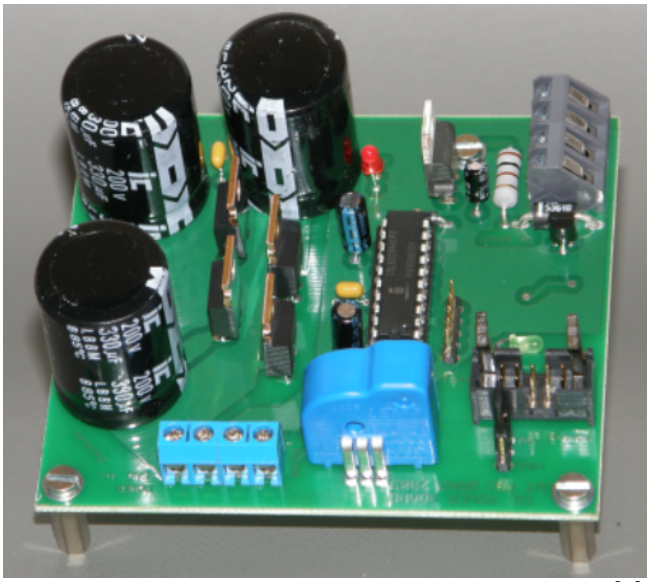


Figure 3.4: ISL power board without heat-sink [6]

Two ISL power boards were present on car4 at the time of this revision meaning that only two of car4's DC motors could be controlled. Originally four of these units were present on car4 but two of them were damaged in the past and have since been

dismantled. The present PCBs were in a good shape. Additional H-bridges were needed to control all four DC motors present on car4. This meant that a new design of H-bridge units that would work in conjunction with the new MCU was needed.

3.4 Summary of analysis

In order to better understand the car4's electronics a block diagram was created (fig. 3.5). The schematic shows all the connected PCBs without the DC motors, servomotors.

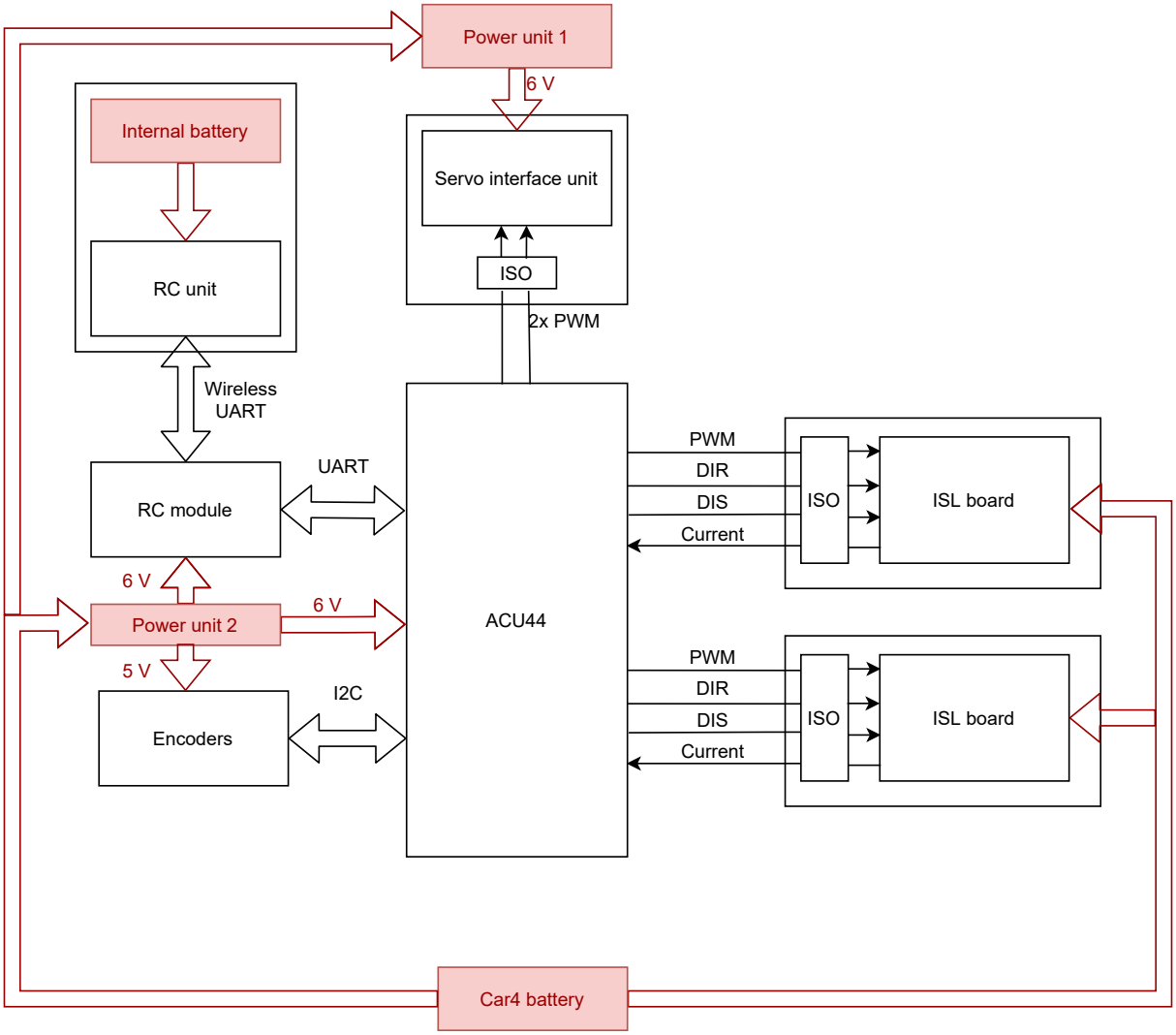


Figure 3.5: Schematic of old car4 electronics

The rotation speed measurement using magnetic encoders AS5048B has been implemented only 2 years before this revision. This system was in good condition and only needed to be connected to some control unit using I2C protocol. The used communication unit that connected the encoders and the MCU was power by a power unit which meant that an additional cable was leading to the I2C connection unit which added complexity to the cable management that could be avoided by powering the unit directly by the MCU.

Single control unit ACU44 served as the MCU. This limited the control possibilities as the ACU44 was capable of controlling only two DC motors and two servomotors. This

in turn meant that the full potential of a 4WS and 4WD could not be utilized.

Two servomotors interface units were present on car4. Single unit was capable of connecting two servomotors so these units impose no limit to the possibility of controlling the steering of all four wheels. But the fact that these units were identical a connection to a single connector on a new MCU would need alternation of either the cable or the connector.

Two H-bridges in the form of ISL power boards were present on car4. This meant that only two DC motors could be controlled. This limited the possibilities for the car4 similarly to the ACU44.

Two power units that served as voltage regulators to reduce the car4's battery to lower operating voltages were present. The first power unit lowered the voltage to $6\,V$ and was dedicated to the servomotors as they could draw substantial current. The second power unit lowered the battery voltage to $12\,V$, $6\,V$, $5\,V$ and $3.3\,V$ levels so that different electronic devices could be powered.

The remote control module was out of scope for this revision and so it was not revised.

The new electronics should became a low-level control algorithm of car4 that controls the core HW and enables addition of higher level algorithms such as LIDAR (= Light Detection and Ranging), infrared sensors, ultrasound sensors, SLAM algorithm (= Simultaneous localization and mapping), trajectory planning, etc. These can be implemented in the future development of car4. Visual representation of this vision can be seen in fig. 3.6.

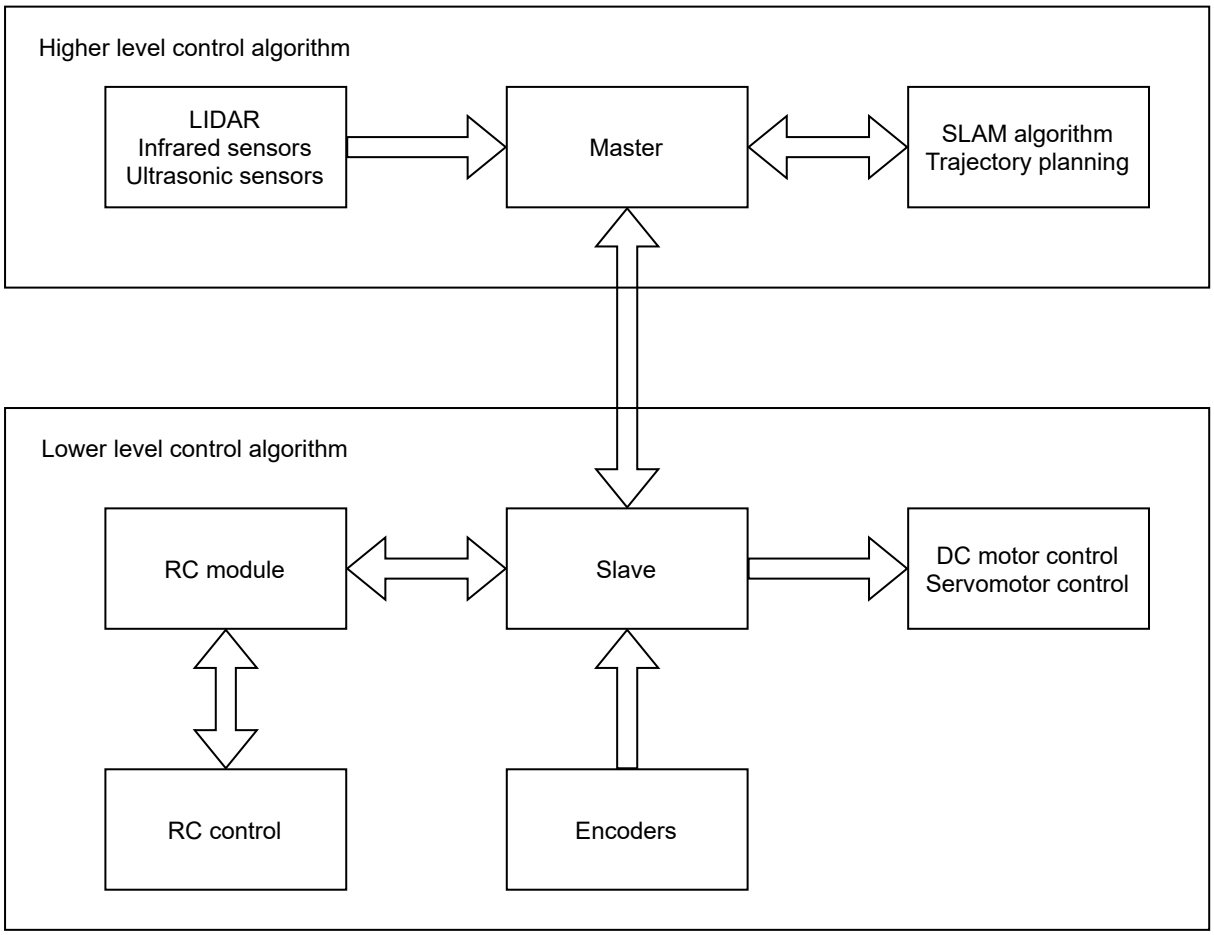


Figure 3.6: Vision of future development of car4

In order to create this lower level control algorithm a list of required changes to the present electronics of car4 was formulated:

- New MCU additionally capable of:
  - Controlling 4 DC motors
  - Controlling 4 servomotors
  - Communication with the speed measuring system whilst providing supply voltage to it
- New H-bridge units capable of:
  - Controlling the speed of the connected DC motor with current control loop
  - Communication with the MCU to receive needed data and send FAULT signal if an error occurs
- New Servo interface unit that would replace the current 2 units

4 Kinematic models

In this chapter the processes of development of kinematic models for car4 were described. Starting from the motivation behind their creation, followed up by their derivation and finally the tests performed with these models. Both models were made and tested using matlab.

4.1 Steering model

4.1.1 Updated model of wheel steering

The steering mechanism used on car4 (fig. 4.1) could be described as a four bar linkage mechanism. The positions of the servomotor's shaft and the axis of rotation of the wheel are fixed in place by the car4's construction.

Figure 4.1: The steering mechanism of car4 with highlighted linkages of 4 bar mechanism

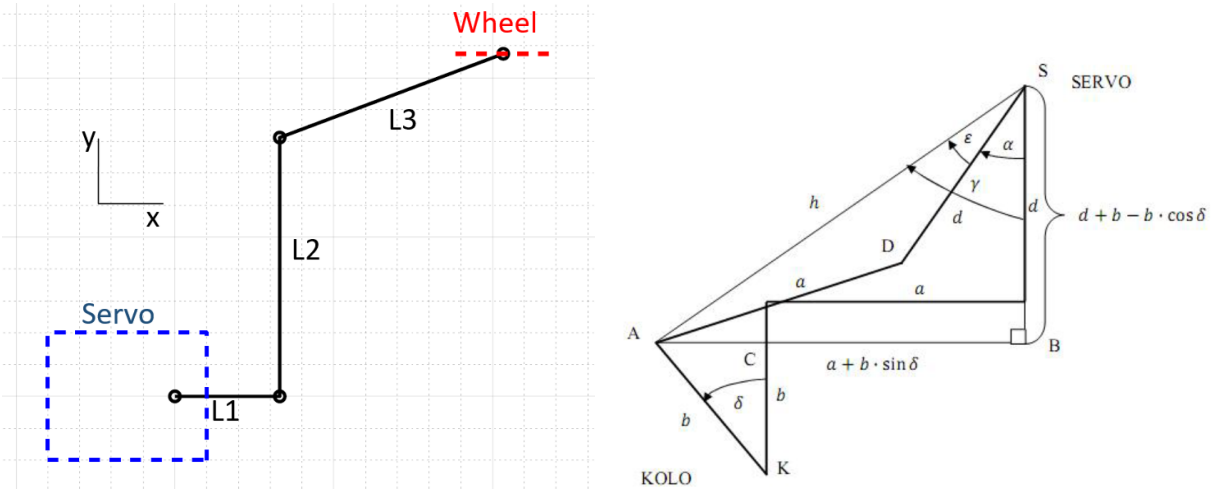
Four bar linkage is a kinematic system with one degree of freedom as seen in equations 4.1 and 4.2. Where i are the degrees of freedom of the system, n is the number of bodies in the mechanism, i_f is the number of available degrees of freedom in 2D space, $\sum \xi$ is the amount of limited parameters of motion and η is the amount of limited parameters of deformation. This means that this mechanism needs only one parameter to

be defined in order to set the positions of all other parts of the mechanism and prevent them from further movement.

$$i = (n-1)i_f - (\sum \xi - \eta) \tag{4.1}$$

$$i = (4-1)3 - (8-0) = 1 \tag{4.2}$$

A model of car4's steering mechanism was proposed in a former thesis [3]. This model (fig. 4.2b) describes the steering mechanism as a four-bar linkage whose setup is determined by the rotation of servo shaft. This model represent a solution to the inverse kinematics of the steering mechanism as the input angle of the model δ is the rotation of the wheel and from which the required rotation of the servo α can then be computed using eqs. 4.3 - 4.6.



(a) Schematic of the new model (b) Schematic of the old model [3]

Figure 4.2: Comparison of steering models

$$h = \sqrt{(d + b - b\cos\delta)^2 + (a + b\sin\delta)^2} \tag{4.3}$$

$$\gamma = \arctan \frac{a + b\sin\delta}{d + b - b\cos\delta} \tag{4.4}$$

$$\epsilon = \arccos \frac{h^2 + d^2 - a^2}{2hd} \tag{4.5}$$

$$\alpha = -\gamma + \epsilon \tag{4.6}$$

This proposed model of car4's steering mechanism was deemed insufficient for the application with Ackerman geometry, as the Ackerman geometry requires precise control of the vehicle's wheels. New model of steering mechanism was created. The new model also describes the steering mechanism as a four bar linkage and works by defining a leading angle at one of the fixed rotary constraints and then computing the positions of the rotary constraints that aren't fixed in position by the construction. Using these computed positions together with the known position of the other fixed rotary constraint the angle of the last bar in the kinematic chain is computed. This way our model can

compute both the wheel rotation based on the servomotor's shaft's angular position or the needed servomotors shaft rotation in order to achieve the required wheel's rotation.

The main difference between the models can be seen when the schematic of the older model (fig. 4.2b) is compared with the schematic of the newer model (fig. 4.2a). The former proposed model presumed that the wheel is facing forward when the links leading from the servomotor's shaft and the wheel's rotary constraint are parallel. Which is not true as can be seen in figure 4.1.

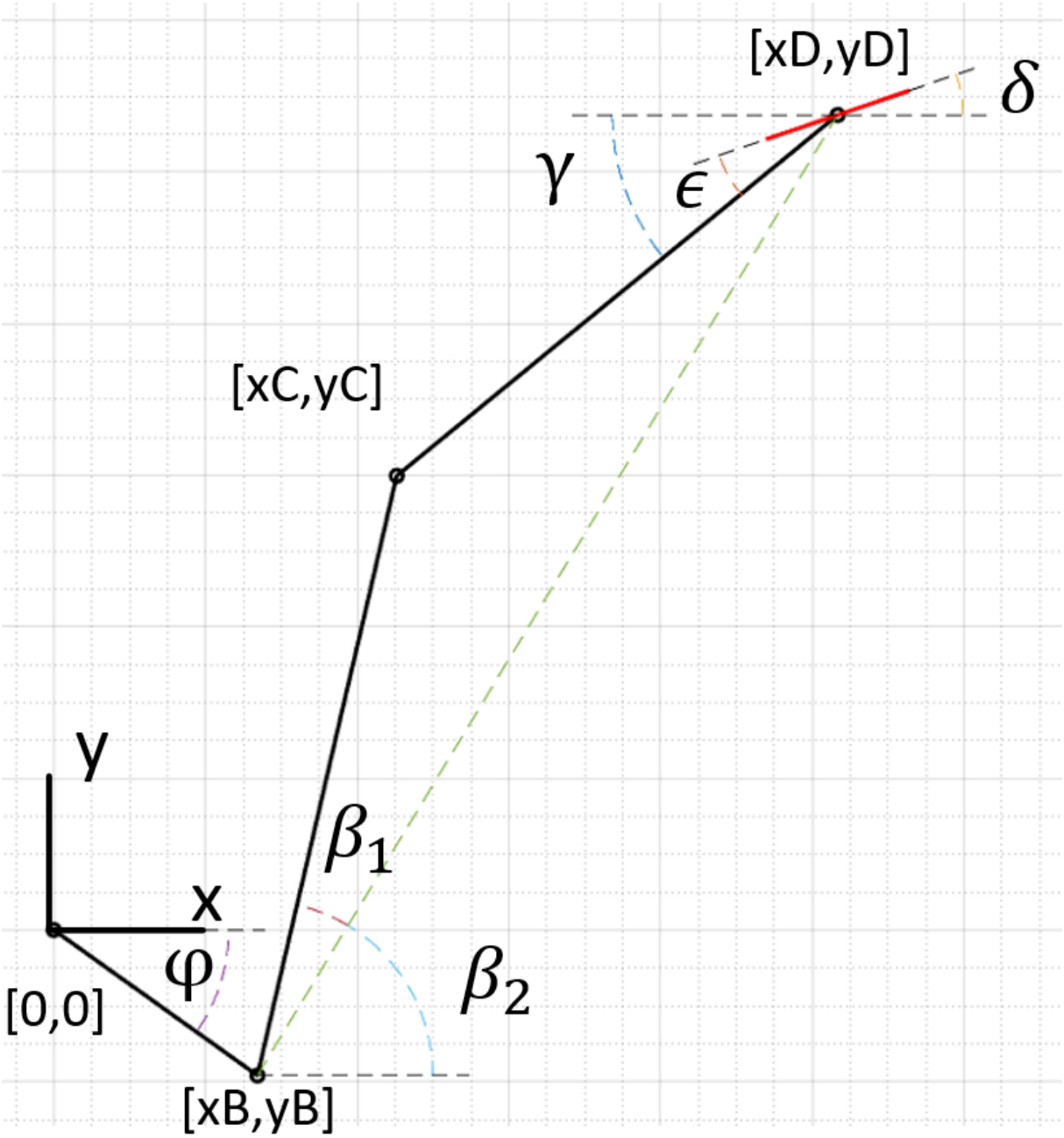


Figure 4.3: Needed angeles for computation of the new steering model

The schematic of the model set to compute the wheel's rotation based on the position of servomotor's shaft with required variables needed for the computation can be seen in fig. 4.3. In the first step position of point B (which represents the position of the

first non-fixed rotary constraint) is computed using the input angle φ (eqs. 4.7 and 4.8). Coordinates of point C (which represents the second non-fixed rotary constraint) are computed similarly using angle β (eqs. 4.13 and 4.14). Where angle β is obtained by adding together angle β_1 computed using cosine law with triangle BCD and angle β_2 is computed with trigonometric function *arcus tangens* (eqs. 4.10 - 4.12). The angle δ is computed by subtracting the constant angle ϵ from angle γ (eq. 4.17). The angle ϵ is a constant angle between the wheel and link L_3 (eq. 4.16, fig. 4.2a). And the angle γ is computed again using trigonomic function *arcus tangens* (eq. 4.15).

$$xB = L_1 \cos\varphi \tag{4.7}$$

$$yB = L_1 \sin\varphi \tag{4.8}$$

$$BD = \sqrt{(xB - xD)^2 + (yB - yD)^2} \tag{4.9}$$

$$\beta_1 = \arccos \frac{L_2^2 + BD^2 - L_3^2}{2 \cdot L_2 \cdot BD} \tag{4.10}$$

$$\beta_2 = \arctan \frac{yD - yB}{xD - xB} \tag{4.11}$$

$$\beta = \beta_1 + \beta_2 \tag{4.12}$$

$$xC = xB + L_2 \cos\beta \tag{4.13}$$

$$yC = yB + L_2 \sin\beta \tag{4.14}$$

$$\gamma = \arctan \frac{yC - yD}{xC - xD} \tag{4.15}$$

$$\epsilon = \arctan \frac{L_{3y}}{L_{3x}} = 21.09^\circ \tag{4.16}$$

$$\delta = \gamma - \epsilon \tag{4.17}$$

The steering mechanism was inspected. The servomotors' shafts have been set to zero position and the wheels' orientations were checked. Slight variations in direction were discovered so the lengths of middle linkages were adjusted as they are comprised of two ball joints connected together by a shaft that is threaded on both sides.

After the adjustments of the steering mechanism the parameters L_1, L_2, L_{3x} and L_{3y} were retrieved from a provided solidworks model from mechlab. The last parameter L_4 was calculated using Pythagoras' rule. The parameters L_1 and L_2 were then measured on car4 itself to double check the values. The parameter L_3 was too difficult to measure without removing it from the construction. But since the parameters L_1 and L_2 corresponded to the solidworks model it has been assumed that the third parameter would also correspond with the model. The mechanism parameters can be viewed in table 4.1.

Arm	L_1	L_2	L_{3y}	L_{3x}	L_3
Length [mm]	33	80	27	70	75

Table 4.1: Parameters of steering mechanism

4.1.2 Simulations of new wheel steering mechanism model

The updated model was used to determine all possible setups of the mechanism. This was done by using the model with servomotor's shaft rotation as input angle (since that is what will be controlled). Values from the range $< 0°; 360° >$ with the step of $0.25°$ were passed through the model. All the output angles were logged. If the input angle results in a physically impossible setup of the mechanism matlab will return an angle with non-zero imaginary part. In order to eliminate these physically impossible setups values with imaginary values were dropped.

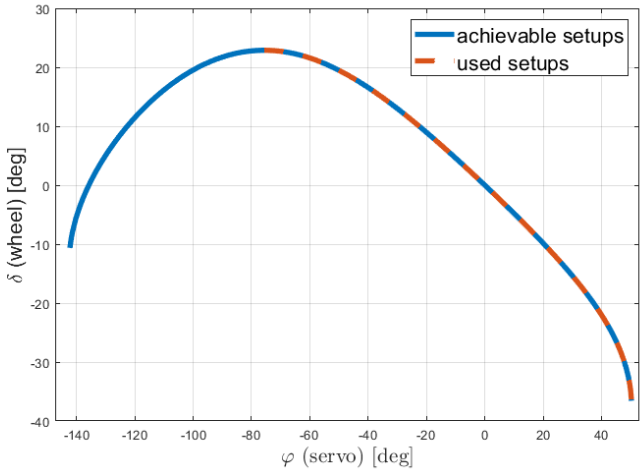


Figure 4.4: All possible and viable setups of mechanism according to the model

The resulting function showed us that this model has one global maximum and two local minimums. In order to control the steering mechanism we were interested only in the interval created by the global maximum and the minimum that contained the value $\varphi = 0°$. Thus the operating interval around the default (zero) rotation of the servomotor's shaft was established. Both the achievable range and the operating range can be seen in figure 4.4.

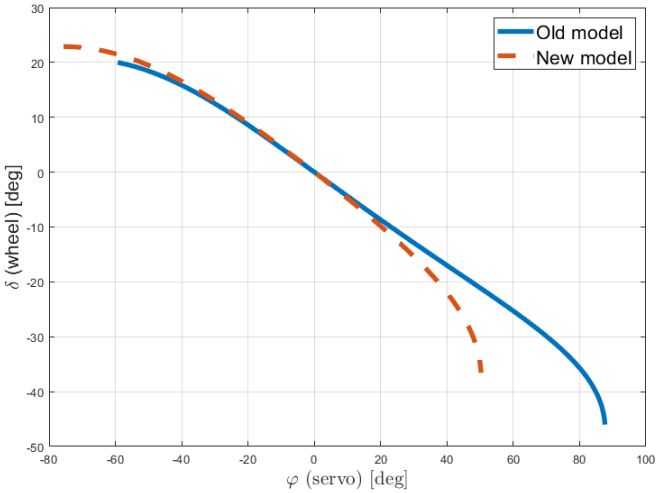


Figure 4.5: Comparison of old and new steering models

The new model of car4's steering mechanism was also compared with the old one. The old model was created using matlab and, similarly to the new model, the operating range was established. The two operating ranges were then plotted against each other to visualize the different behavior (fig. 4.5). The models are similar around the zero position of the servomotor's shaft, but vary substantially the further the servomotor's shaft rotates. Another difference can be observed in the operating ranges as the old model has the operating range of $< -59.3°; 87.6° >$ and the new model has the operating range of $< -75.8°; 50.3° >$.

4.2 4WS model using Ackerman geometry

4.2.1 Created model

In order to implement Ackerman geometry the parameters of car4 were retrieved from [2] and can be seen in table 4.2. Where l is the length of car4, w is the width of car4 and r is the radius of car4's wheels.

Parameters	l	w	r
Length [mm]	500	400	75

Table 4.2: Parameters of car4

The schematic from which the model was derived can be seen in fig. 4.6. In this model it is assumed that the center of mass is in the middle of the vehicle. The model has three inputs and eight outputs:

- Input a - the distance of ICR in x coordinates
- Input R - the distance of ICR in y coordinates
- Input v - the required forward velocity
- Outputs ω_1 to ω_4 - the required wheels' angular velocities
- Outputs δ_1 to δ_4 - the required wheels' rotations

For the computation imaginary front and rear center wheels that comply with the inputs are used. In the first step the required rotation of the imaginary wheels is computed (eqs. 4.18 and 4.19). Then using the law of sines on triangle Re-Fr-ICR the distance of the imaginary rear wheel and the ICR is calculated (eq. 4.20). Using the law of cosines on triangle Re-P-ICR the distance of the vehicle's center of mass and ICR is computed (eq. 4.21).

$$\delta_R = -\arctan\frac{a}{R} \tag{4.18}$$

$$\delta_F = \arctan\frac{l-a}{R} \tag{4.19}$$

$$R_R = l\frac{\sin(90° - \delta_F)}{\sin(\delta_F - \delta_R)} \tag{4.20}$$

$$R_0 = \sqrt{R_R^2 + (a + \frac{l}{2})^2 - R_R(a + \frac{l}{2})\cos(90° + \delta_R)} \tag{4.21}$$

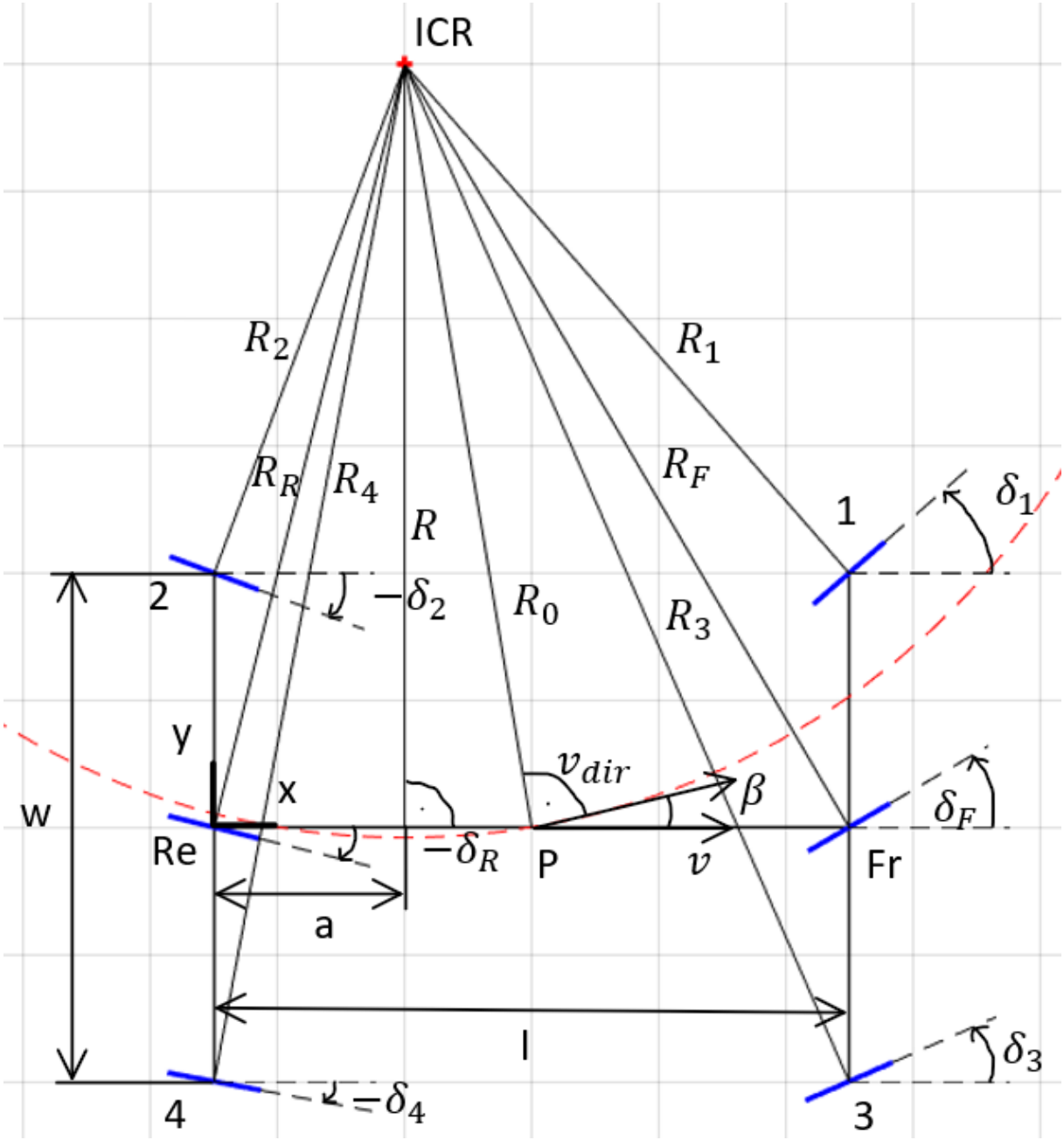


Figure 4.6: Schematic of kinematic model using Ackerman geometry

Next the angle between the input forward velocity v and the velocity v_{dir} acquired by fulfilling the remaining input arguments is computed (eq. 4.22). The value of the velocity v_{dir} can then be computed using eq. 4.23. Using Pythagorean equation the distances of each wheel from the ICR are calculated (eqs. 4.24 - 4.27).

$$\beta = 90° - \arcsin\left(90° + \delta_R\right)\frac{|R_R|}{R_0} \tag{4.22}$$

$$v_{dir} = \frac{v}{|\cos\beta|} \tag{4.23}$$

$$R_1 = \sqrt{(R - \frac{w}{2})^2 + (a + l)^2} \tag{4.24}$$

$$R_2 = \sqrt{(R - \frac{w}{2})^2 + a^2} \tag{4.25}$$

$$R_3 = \sqrt{(R + \frac{w}{2})^2 + (a + l)^2} \tag{4.26}$$

$$R_4 = \sqrt{(R + \frac{w}{2})^2 + a^2} \tag{4.27}$$

By knowing the velocity of the center of mass v_{dir}, the distance between the center of mass and ICR R_0 and the distances of each wheel from the ICR R_1, R_2, R_3 and R_4 the needed velocities to achieve no-slip control can be computed. These velocities can then be divided by the wheel's radius r which results in the required angular velocities of each wheel (eqs. 4.28 - 4.31).

$$\omega_1 = v_{dir} \frac{R_1}{R_0 r} \tag{4.28}$$

$$\omega_2 = v_{dir} \frac{R_2}{R_0 r} \tag{4.29}$$

$$\omega_3 = v_{dir} \frac{R_3}{R_0 r} \tag{4.30}$$

$$\omega_4 = v_{dir} \frac{R_4}{R_0 r} \tag{4.31}$$

All the required angles of each wheels' rotations can be computed using the trigonometric function *arcus tangens* (eqs. 4.32 - 4.35).

$$\delta_1 = \arctan \frac{l - a}{R - \frac{w}{2}} \tag{4.32}$$

$$\delta_2 = -\arctan \frac{a}{R - \frac{w}{2}} \tag{4.33}$$

$$\delta_3 = \arctan \frac{l - a}{R + \frac{w}{2}} \tag{4.34}$$

$$\delta_4 = -\arctan \frac{a}{R + \frac{w}{2}} \tag{4.35}$$

This model hits a singularity when the input argument R is infinity (vehicle is not turning). When this happens the velocity of each wheel is the same and is computed using equation 4.36. The model computes the wheels' rotations correctly when $R = \infty$.

$$\omega = \frac{v}{r} \tag{4.36}$$

4.2.2 Performed simulations

In order to test the created model a function in matlab was created that allows to plot the model. The model is capable of plotting the ICR, the future trajectory if the inputs were to stay the same, wheels' rotation and perpendicular lines to each wheel. The plotted

model with input parameters $a = 0.1$, $R = 0.6$ and $v = 10$ can be seen in figure 4.7.

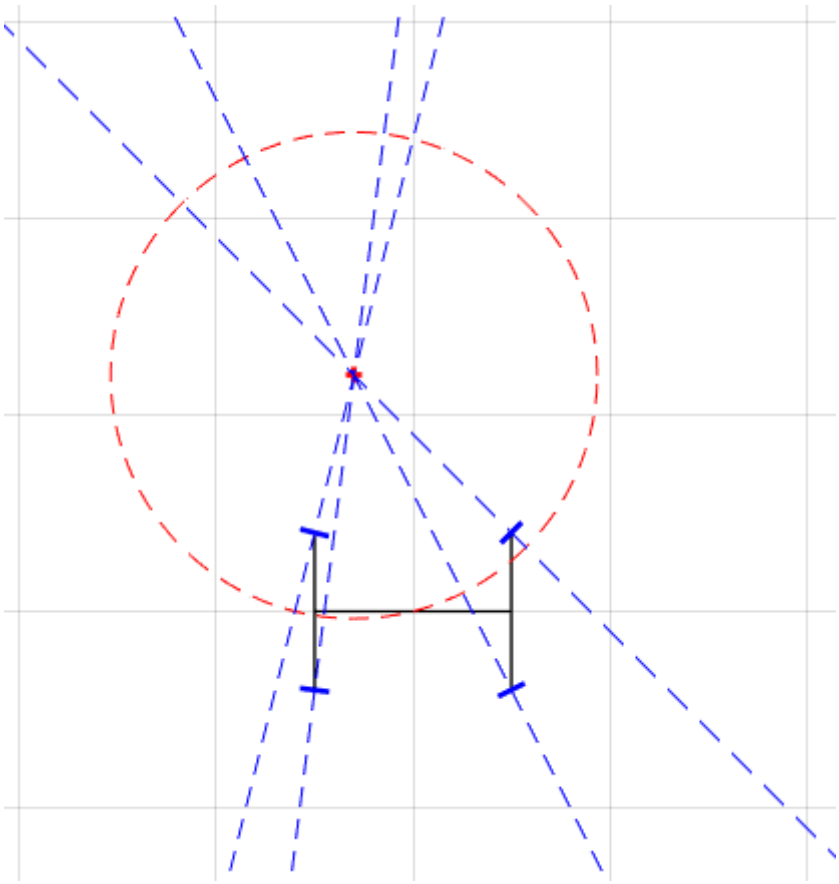


Figure 4.7: Ackerman model output

This way the wheels' rotation angles δ_1 - δ_4 were checked visually using different sets of input data. The used testing input data sets placed the ICR in various positions that could be divided into:

- Behind the rear axle, left side of the vehicle
- Between the rear axle and the front axle, left side of the vehicle
- In front of the front axle, left side of the vehicle
- Behind the rear axle, right side of the vehicle
- Between the rear axle and the front axle, right side of the vehicle
- In front of the front axle, right side of the vehicle

For all of these positions the model worked as inteded.

The computed angular velocities cannot be checked this way. In order to check the velocities velocities for several different setups were calculated without the model and then the results were compared with results from the model.

5 New car4 electronics

This chapter is divided into three parts. The initial theoretical design of car4's new electronics. Afterwards the designing processes of new electronics. And finally the implementation of new electronics onto car4.

5.1 Design of new car4 electronics

After evaluating the information gained from the revision of car4's electronics described in the chapter 3 an updated design of the electronics was proposed and can be seen in fig. 5.1.

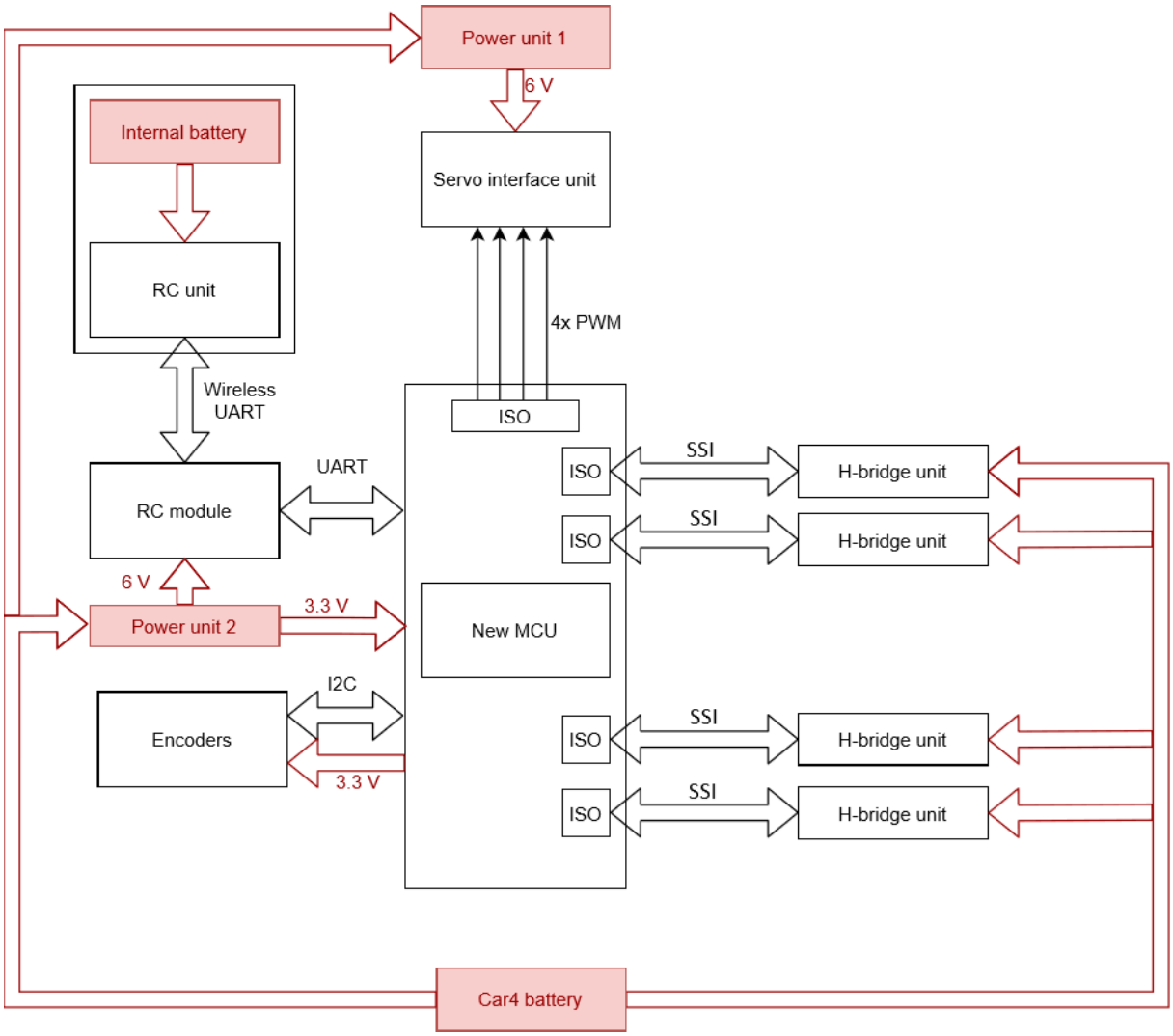


Figure 5.1: Schematic of designed updated car4 electronics

In the new design there is a single MCU that handles data from sensors, communicates with the RC module and computes required angular velocities and rotations of all wheels based on the input from the RC module. It then generates needed PWM signals to control the servomotors based on computed required wheels' rotations and sends them to a new servo interface unit to which all servomotors are connected. It also sends the required speed to four new H-bridge PCBs which will all have a microcontoller on it's own that will allow them to control the speed of the connected motor and thus free up computation power for the MCU. In order to achieve this state given the former schematic (fig. 3.5) several changes were needed.

For the **new control unit** following requirements had been formulated:

- Microcontroller capable of controlling the whole car4
- Programming and debugging port
- UART connector for communication with the RC module
- USB connector for PC connection
- 2x I2C connectors for Encoders and Battery control module
- Single connector to send 4 PWM signals to control servomotors, galvanically isolated
- 4x connectors each for one H-bridge using using some communication, galvanically isolated
- Power connector for $3.3\,V$ supply
- LEDs for visual control of powered on PCB

The requirements for the **new servomotor interface unit** were also formulated:

- Connector for PWN control signals from the MCU
- 4x connectors for servomotors
- Voltage regulator for the galvanic isolator on the MCU
- Power connector for $6\,V$ supply

And the requirements for the **new H-bridge PCBs** were established:

- Microcontroller for communication with the MCU and to control the connected DC motor
- Programming and debugging port
- H-bridge in an IC
- Current gauge
- Voltage regulators for the microcontroller, H-bridge IC, current gauge and the galvanic isolator on the MCU

- Connector for the input from the MCU
- Power connector for the battery
- DC motor connector
- LEDs for visual control of powered on PCB

5.2 Designs of new PCBs

5.2.1 MCU design

As the base for the design a thesis [4] that focused on designing a new MCU for the car4 was used. After careful inspection several undesirable design features were discovered, the size of the PCB was too large to be installed onto the car4 and the design didn't meet the requirements stated in section 5.1. The produced PCBs with all the components mounted can be seen in figure 5.2.

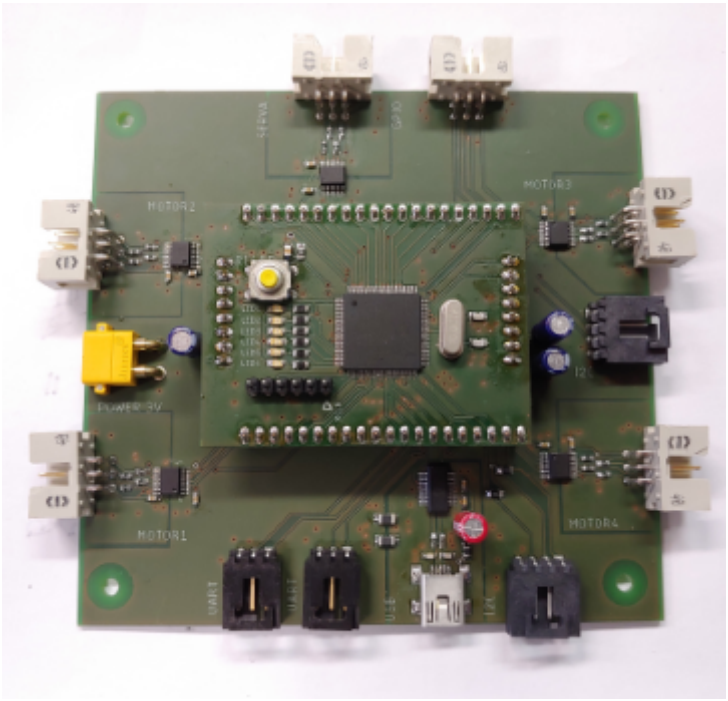


Figure 5.2: MCU with all components mounted

The ports were replaced in order to fulfil the requirements. Additionally a UART port for future use was added. Galvanic isolator from the port containing 4 GPIOs was removed as the used types supported only one way communication which would reduce the intended general use. During the design phase of new electronics it has been decided that H-bridge units will not be made. The required signals to control the current ISL boards of high-speed PWM and digital IOs DIR and DIS were brought to the motor control ports.

The connectors used in the original design were replaced by by two differen types of connectors. One for the H-bridge PCB connection, servomotor unit connection and GPIO port. And anotrher for the UART and I2C ports. This change was performed because contacts for the former connectors were difficult to crimp.

The LEDs were replaced as their datasheet stated forward current ($20\,mA$) was above the microcontroller's stated input limit ($10\,mA$). The forward current of the new LEDs is $2\,mA$. In order to achieve this state the required resistor was determined using eq. 5.1. Where V_{cc} is the supply voltage, V_f is the forward voltage of used LED and I_f is the forward current of used LED.

$$R = \frac{V_{cc} - V_f}{I_f} = \frac{3.3\,V - 1.8\,V}{2\,mA} = 750\Omega \tag{5.1}$$

The microcontroller *dsPIC33EP512MU810* used in the original design has been kept. This microcontroller contains many reamappable pins which allows to easily alter the design if the need arises. It also contains 4 UART modules, 4 SPI modules and 2 I2C modules so the requirements for communications are satisfied.

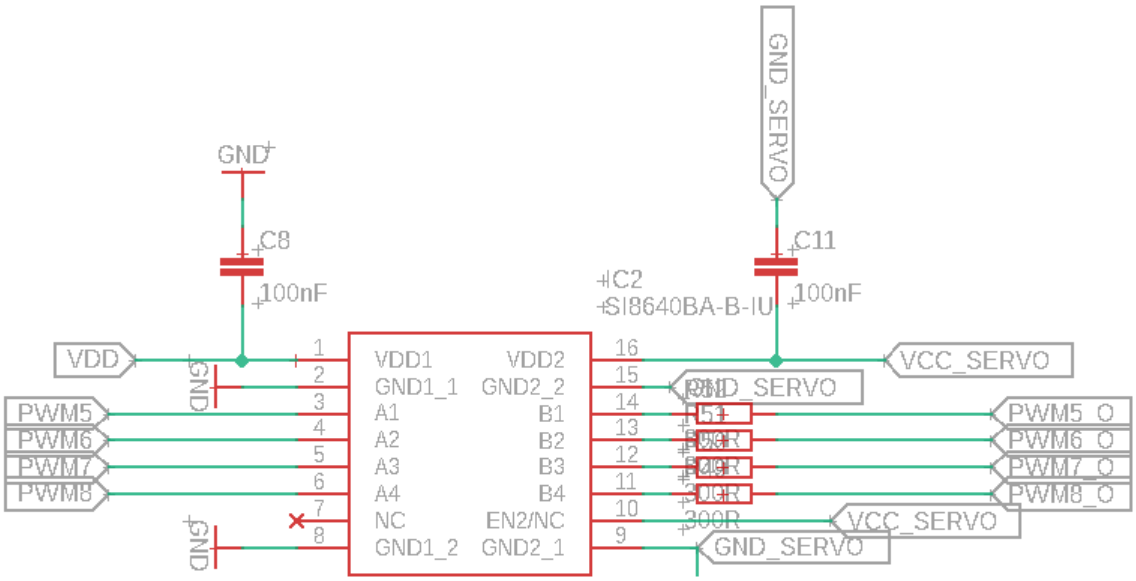


Figure 5.3: Schematic of digital isolator for servomotor interface unit

The digital isolators *Si864x* used in the original design were kept. These offer up to $150\,Mbps$ and have wide operating supply voltage $2.5 - 5.5\,V$. The setup of the isolators was altered in order to comply with the recommended one stated in the manufacturer's datasheet [18]. The schematic connection of a digital isolator can be seen in fig. 5.3 (in this case it is the port for the servomotor interface unit).

For the USB connector the *FT231X* chip was used just like in the original design. Using this device a UART module can be used to establish a USB port. For proper function two ferrite beads were added to separate the ground of the *FT231X* chip from the rest of the MCU's ground. The schematic connection of this device can be seen in 5.4.

Capacitors were added along the power lines in order to stabilize them. During the design of the new MCU for the car4 all the unused routes were deleted and all connectors moved around in order to reduce the size of the MCU. In the end the lower PCB with the connectors was reduced by the factor of 0.611 (from $130\,mm \times 140\,mm$ to $103\,mm \times 108\,mm$) and the top PCB with the microcontroller was reduced by the factor of 0.436 (from $86\,mm \times 68\,mm$ to $58\,mm \times 44\,mm$).

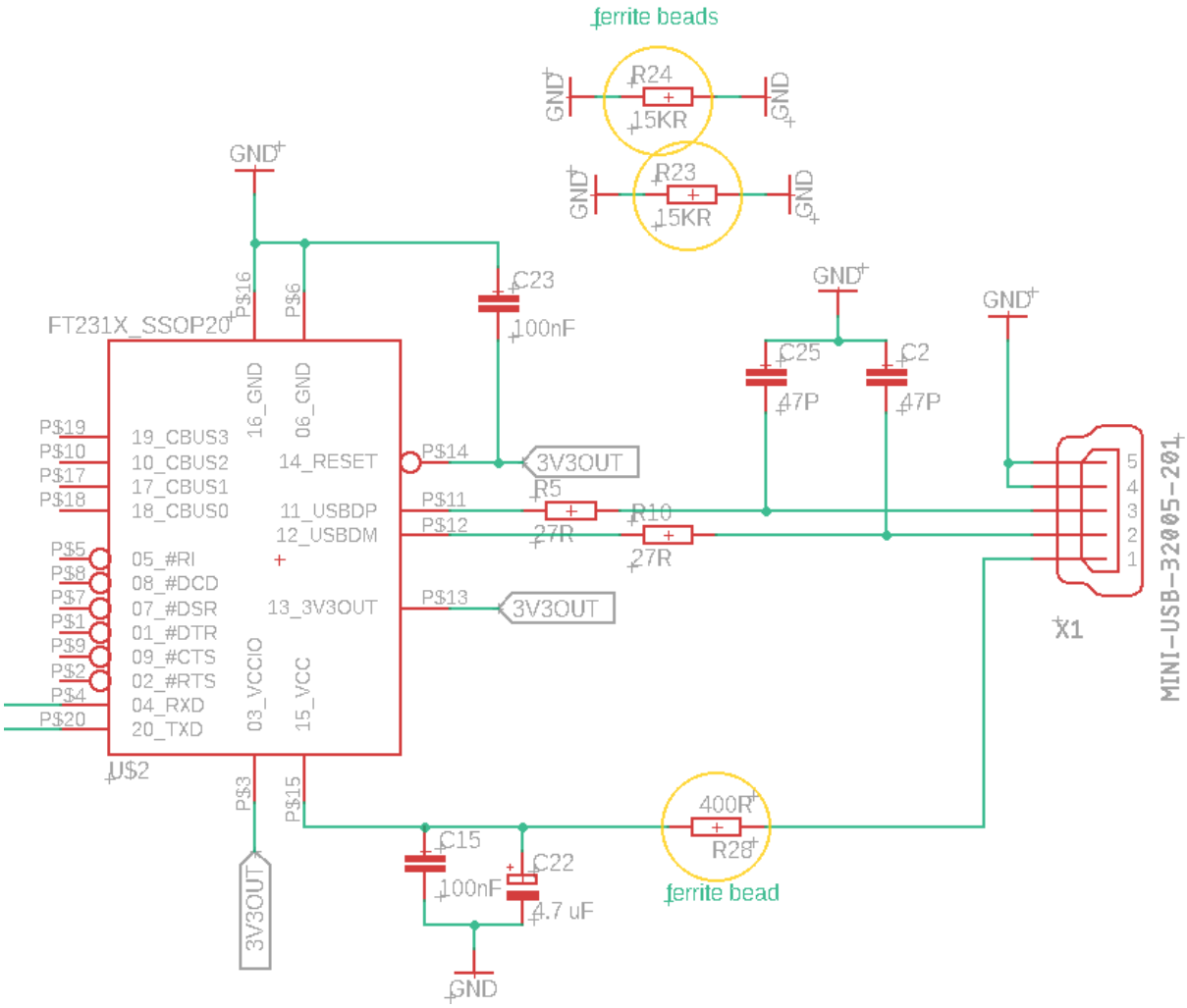


Figure 5.4: FTDI connection schematic

A list of the most noticeable improvements of the new MCU over the old *ACU44* are listed here:

- Increase of Program Flash Memory from 128 kB to 536 kB
- Increase of RAM from 16 kB to 52 kB
- Increased number of DC motors that can be driven from 2 to 4
- Increased number of servomotors that can be driven from 2 to 4
- Increase the number of I2C ports from 1 to 2

5.2.2 SERVO interface unit design

New PCB which complies with the requirements for the servo interface unit stated in section 5.1 was designed and manufactured. The produced PCB with all the components mounted can be seen in figure 5.5 and the schematic of the final design can be seen in figure 5.6. The type of the connector was already determined during the design of the MCU so it is obvious the same was also used on this PCB.

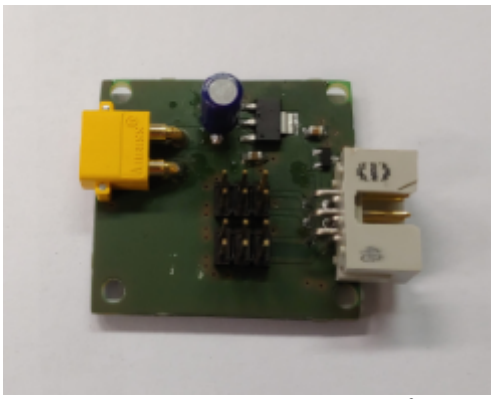


Figure 5.5: New servomotor interface unit

The servomotors require only basic pin header with contacts' pitch of 2.54 mm. Two 2x3 pin headers with the required contacts pitch were used as connectors. In order to drop the input voltage from 6 V to 3.3 V for the digital isolator the voltage regulator *LMS8117A* was used.

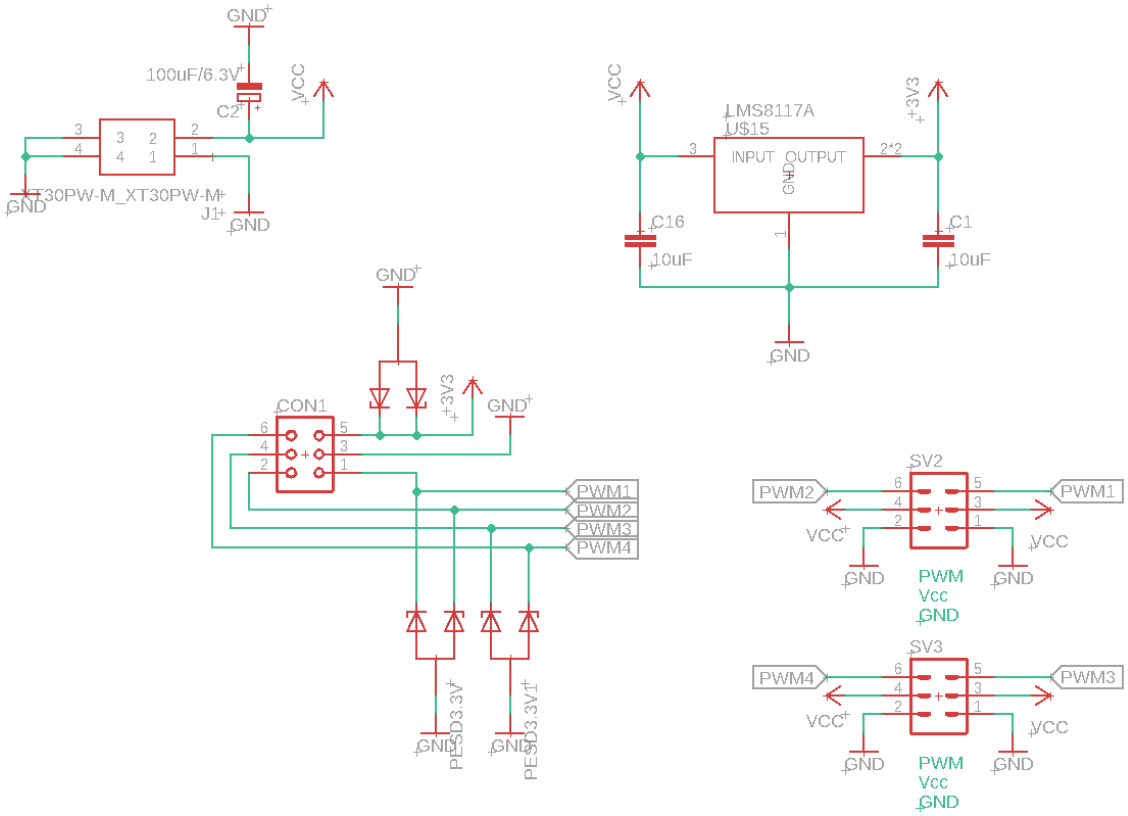


Figure 5.6: Servomotor interface unit schematic

5.2.3 H-bridge PCB design

For the new H-bridge PCB, H-bridge board used for the "Hummer" project [1] served as inspiration because it meets all of the formulated requirements and was field tested.

During the design phase of this PCB it has became apparent that the complexity of mounting and testing four of these units would most likely hinder the revision as a whole. Thus the decision to drop the manufacturing process of these units was made.

5.3 Implementation

Since not all of the needed PCBs for the new updated car4's electronic design (fig. 5.1) were produced several adjustments needed to be made in order to install the updated electronics. The state of the electronics after the installation of the new PCBs onto the car4 and the needed alterations done can be seen in figure 5.7.

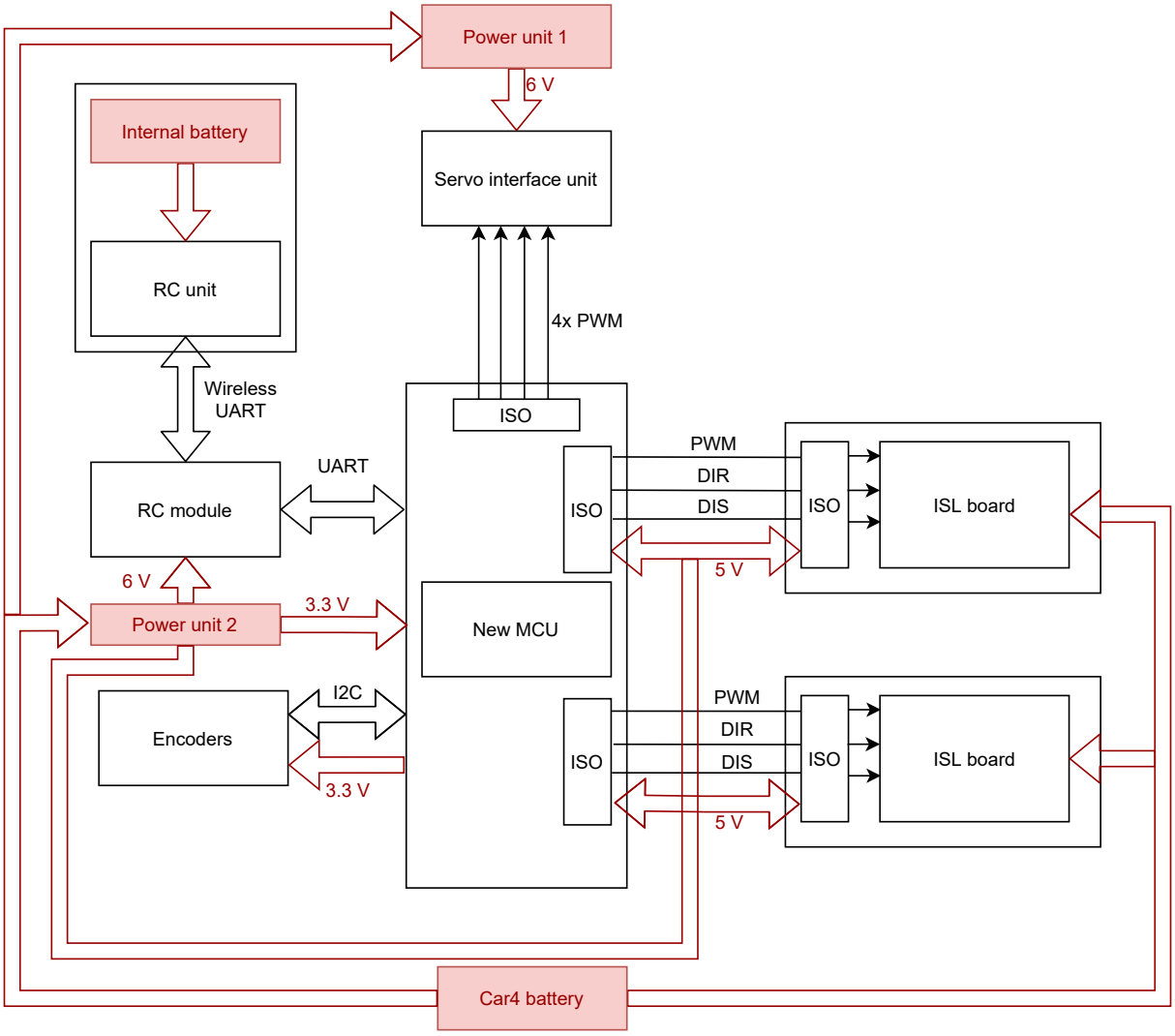


Figure 5.7: Schematic of implemented car4 electronics

The main difference was the control of car4's DC motors. Only two ISL boards that serve as H-bridges each capable of controlling a single DC motors were present on car4. Additionally in the new design of car4's electronics the digital isolators were present on the MCU. This meant that both the MCU and the ISL board present on car 4 had galvanic isolators that needed to be powered by both sides. $5\,V$ supply from one of the car4's power units was connected to both digital isolators so that the communication between the MCU and the ISL boards was possible. The ISL board uses connectors that are similar to the ones used on the new MCU except that the ISL board uses ones with 10 contacts whilst the new MCU uses one with 6 contacts. The contacts of the ISL board's connector were manually aligned so that the input signals from the MCU were connected where needed.

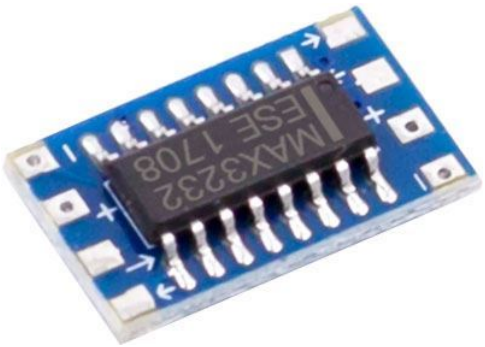


Figure 5.8: TTL - RS232 converter using MAX3232 [21]

During the testing of the new hardware it has been discovered that a TTL to rs232 converter is required in order for the MCU to communicate with the RC module. A small PCB with *MAX3232* module (fig. 5.8) was placed in between the MCU and the RC module which solved the problem and rectified the mistake.

The selected microcontroller *dsPIC33EP512MU810* is $5\,V$ input tolerant but outputs signals with $3.3\,V$ logic. The magnetic encoders were set to work with $5\,V$ logic level. In order to avoid any potential problems that might have appeared because of the different logic levels the PCBs with the magnetic encoders *AS5048B* were altered. This change didn't require any major changes only two neighbouring pins of the IC package of the magnetic encoder needed to be connected.

6 New control algorithm

In this chapter the process of implementing the new control algorithm is described. This chapter is further divided into four sections. Three sections are concerned with the speed controller, wheel steering controller and the Ackerman geometry. Each section describes the implementation of it's solution. This includes reasons why the certain method of implementation was chosen and the tests performed in order to confirm their functionality.

As the base of our control algorithm software developed in simulink in thesis [4] was used. The simulink model used "MPLAB Device Blocks for Simulink" library which contains many pre-programmed blocks that allow much easier implementation of different features such as microcontroller setup, I2C and UART communication, Output compare and PWM setup, and many others. Based on the created simulink model C code was generated and then loaded into the microcontroller.

6.1 Wheel rotation speed control

One of the requirements of Ackerman geometry is the ability to control the speed of the vehicles wheels. In our case the wheels were connected to the shaft of the motor with a 1:14 gearbox. For the measuring of the wheels' speed magnetic encoders described in section 3.1 were used. These encoders transfer the measured angular position of the motor's shaft φ_k using I2C protocol with 14 bit resolution. Using "BUS I2C MASTER" block in simulink the bitrate of the I2C communication was raised from $124.89\,kHz$ to $343.10\,kHz$ as it was too slow and some values were lost.

The angular position of the motor's shaft is then differentiated which gives us the angular velocity of the motor's shaft ω_k (eq. 6.1) and afterwards checked for the possible occurrence of integer overflow.

$$\omega_k = (\varphi_k - \varphi_{k-1}) f_s \tag{6.1}$$

Integer overflow happens when an operation tries to increase (or decrease) the value of a integer variable past it's range. In our case 14 bits can be arranged to represent 16,384 different values so we have an interval of positive whole numbers $< 0; 16383 >$. If something were to attempt to raise the value 16383 by 1 the value would become 0. And similarly the value would become 16383 if it were to decrease by 1 from 0.

The implemented overflow safety works by checking if the differentiated speed isn't greater than half of the 14 bit resolution or smaller then the negative half of the 14 bit resolution. The control algorithm has the sampling frequency of $f_s = 1\,kHz$. With this value we can calculate the maximum theoretical angular velocity that the magnetic encoder could reliably differentiate (eq. 6.2).

$$\omega_{max} = 2\pi f_s \frac{60}{2\pi} = 60000\,rpm \tag{6.2}$$

The nominal speed of the used DC motor *PD4266-24* is 5900 rpm [20] which means that we can safely compute the angular velocity ω_k using eq. 6.1 and use the above mentioned overflow safety without it failing due to overflow occurring by the sheer speed of the motor.

Each position of the magnetic encoder's 14 bit resolution could be interpreted as a *tick*. The equation 6.1 would then be in $\frac{ticks}{s}$. This type of units was deemed undesirable and using gain transformed into the standard $\frac{rad}{s}$.

In order to reduce the noise measured by the sensors a moving average filter using the last ten values was added. Acquired data from the encoders and the output from the moving average filter can be seen in fig. 6.1.

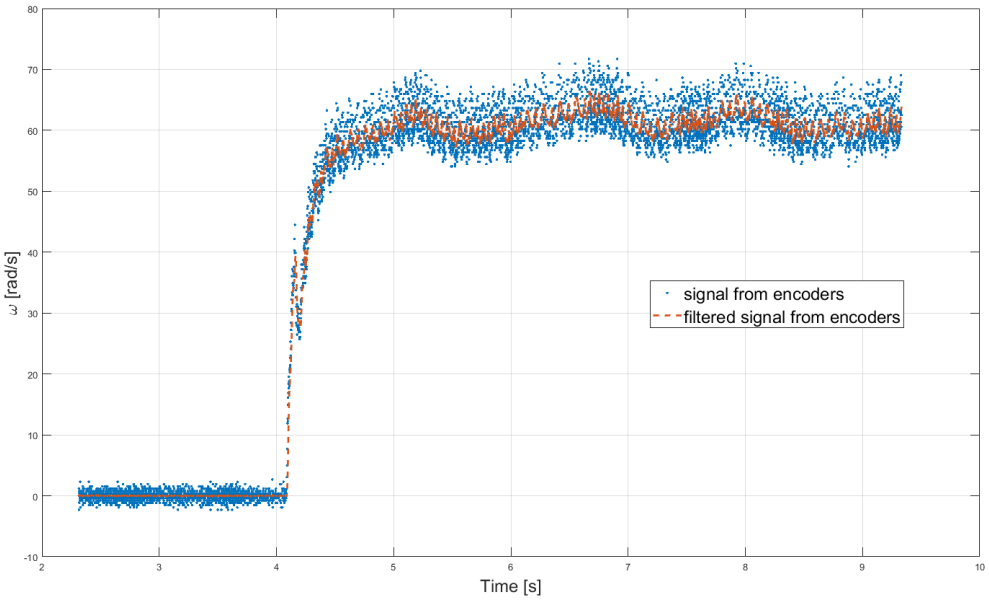


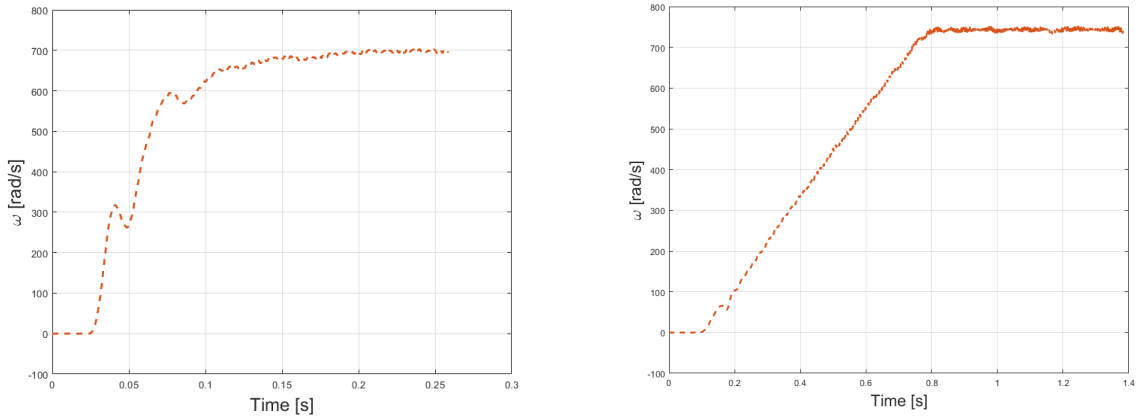
Figure 6.1: Signal from encoder

During the testing of different speed controllers a strange behaviour of the motors was discovered. When any of the car4's DC motor is connected directly to the battery it doesn't behave as expected following an exponential curve. Instead the motor's speed drops twice during the acceleration as can be seen in fig. 6.2a.

For comparison the DC motors were connected directly to a laboratory power supply the transient event can be seen in fig. 6.2b. The behaviour again doesn't follow an exponential curve but in this case a linear curve. The voltage generator has a built-in current limiter that started working shortly after providing power to the motor and caused the described event.

Since the motors behave predictably when connected to a voltage generator and unpredictably when connected directly to the battery. It was assumed that the old battery packs were the source of the disturbance. The use of some type of PID controller was decided. This type of controller would work well for the purpose of speed control.

All of the tested PID controllers have been implemented using "Discrete PID Controller" block in simulink. This block allows to easily set each gain of the controller and additionally allows the use of many other features. The features used in our case were



(a) Connected to battery (b) Connected to voltage generator

Figure 6.2: Measured speed during transient events

output saturation to limit the output to $< -100; 100 >$ (duty cycle of PWM signal to ISL board) and the sample time of the block set to $10\,ms = 100\,Hz$. The sample time was set 10x slower that then the sample time of the whole algorithm in order to prevent measured values used by one iteration of the controller to affect the following iteration.

After manually testing different types of PID controllers a PI controller with the values in table 6.1 was determined to have best behaviour. The behaviour of the two front DC motors controlled by this PI controller can be seen in figures 6.3 and 6.4.

PI parameters	value [-]
K_P	0.2
K_I	2

Table 6.1: Parameters of used PI controller

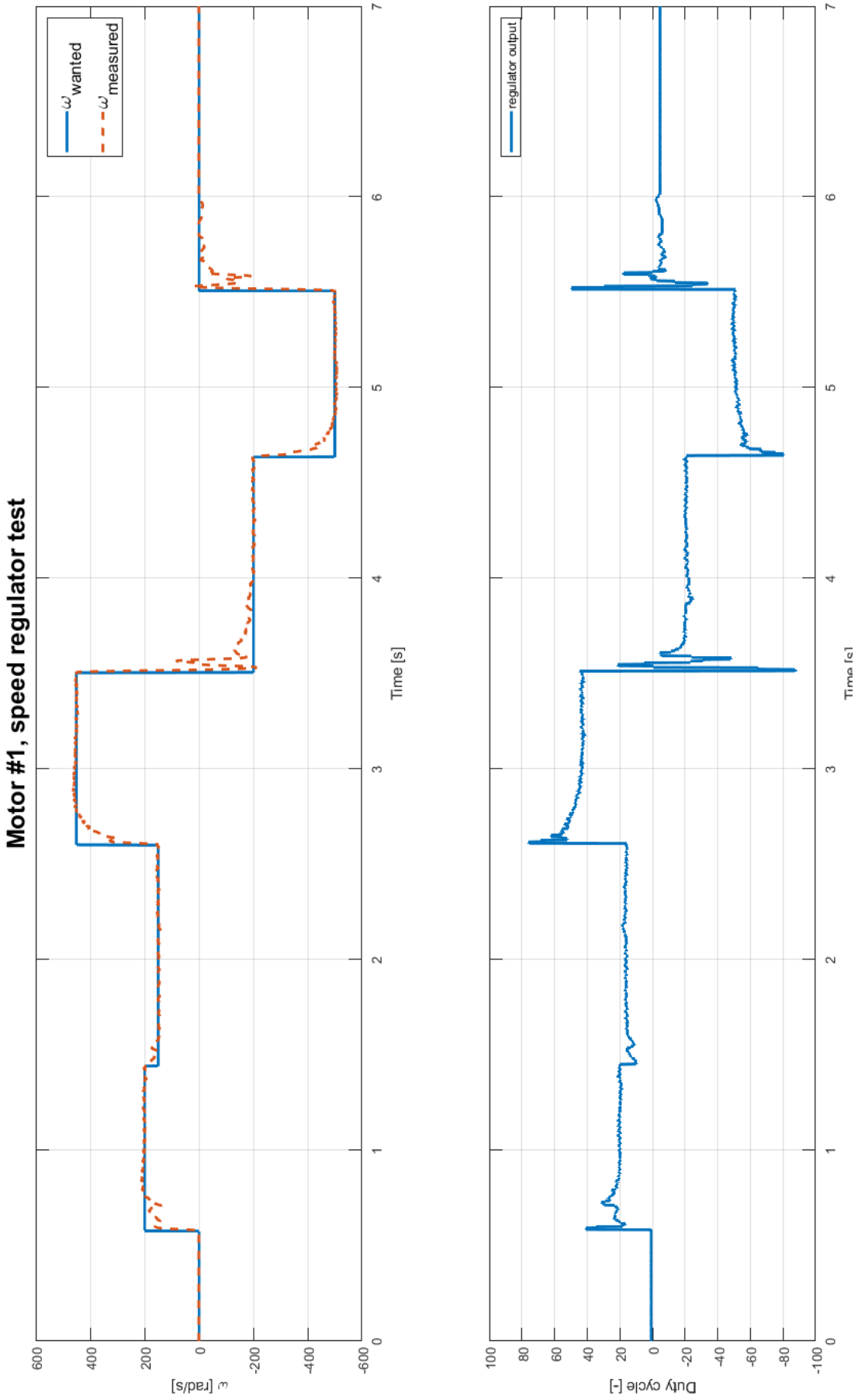


Figure 6.3: Speed regulator test of motor 1

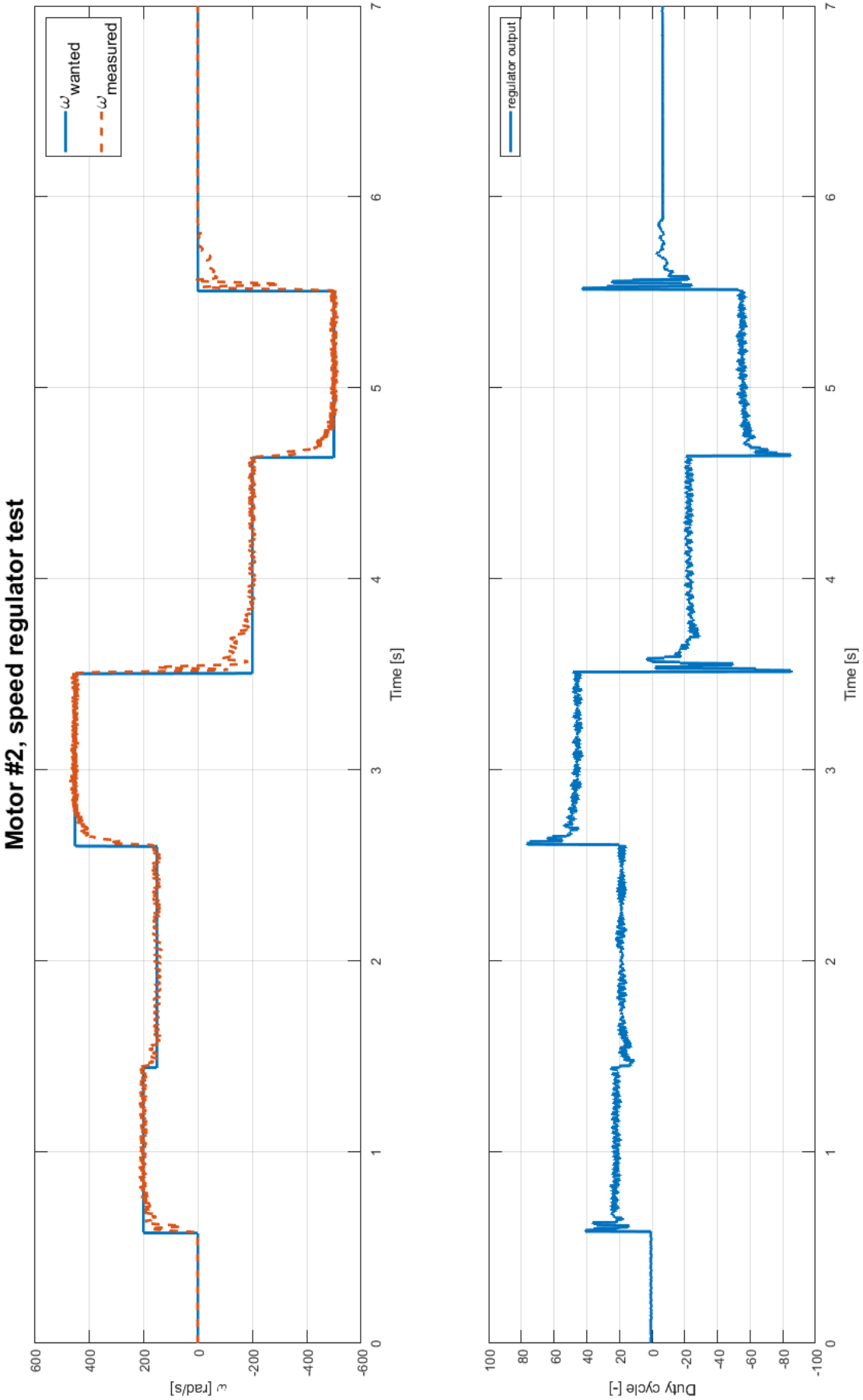


Figure 6.4: Speed regulator test of motor 2

6.2 Wheel steering control

Shortly after implementing the model described in section 4.1 using a look-up table it has been observed that the steering mechanism behaves differently than anticipated. The actual steering range was therefore measured and discovered to be $< -30°; 29° >$. This meant that both models were inapplicable since their operating ranges were very different as can be seen in fig. 4.5. The problem was analyzed and the most likely cause of this difference was determined to be the way the rotary constraint of the wheel is constructed. It is constructed using two ball joints that aren't on the same axis perpendicular to the XY plane of the vehicle as both steering models presume. The angle between the perpendicular axis and the axis going through both of the ball joints was measured using solidworks model provided by mechlab. The axis is rotated by 7.7°.

Since the steering mechanism can not be described in 2D plane a different approach in modeling was required. Options using first generation of simscape or altering the provided solidworks model were consider but deemed too time consuming. Instead a set of measurements was performed on the steering mechanism using a protractor. The setup can be seen in figure 6.5. Unfortunately the COVID pandemic restricted our options so this DIY solution was implemented.

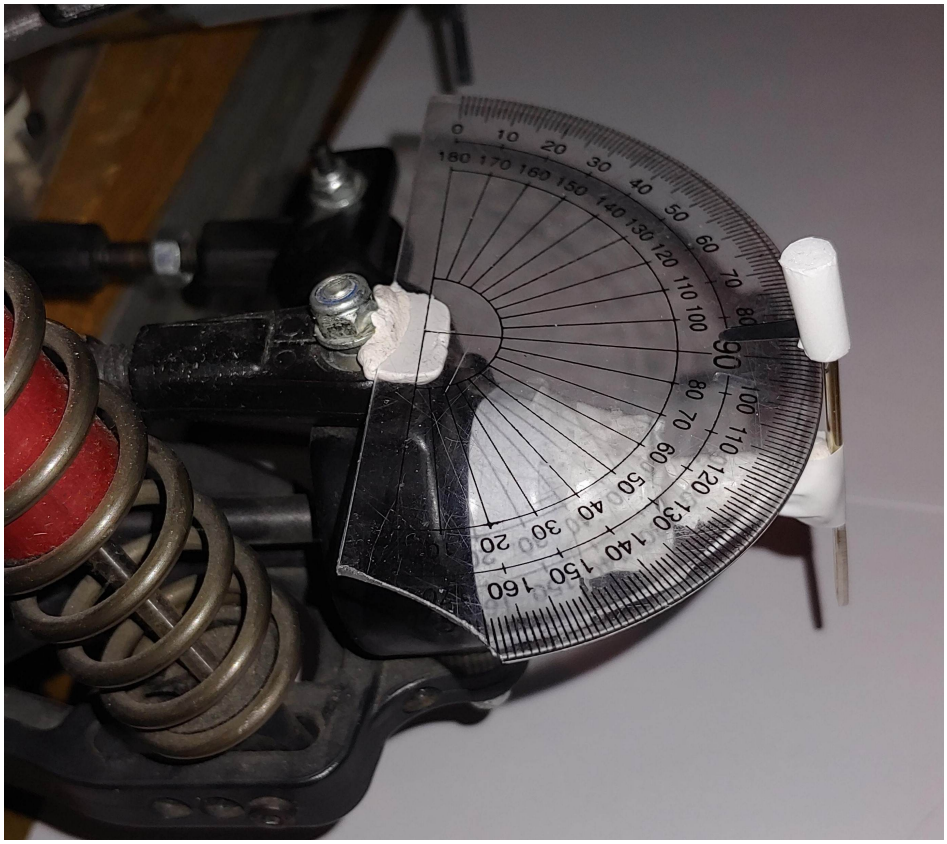


Figure 6.5: Measuring setup of wheel steering mechanism

The measured data were than paired with the inputs of each measurement in order to form dataset containing 33 values (fig. 6.6) for a 1D look-up table which was then implemented in the control algorithm. In order to save computing power of the MCU a single 1D look-up table was implemented with data acquired by measuring the front left steering mechanism's behaviour. The same mechanism only rotated along the the x axis

of the vehicle is employed on the right side. The data for the right steering mechanism can then be gained by multiplying the input to the look-up table by -1 as the range of the right steering mechanism is swapped (to $< -29°; 30° >$) and then multiplying the output of the look-up table by -1 as the servo must turn the other way around to achieve the required angle. The implemented 1D look-up table can be seen in fig. 6.7.

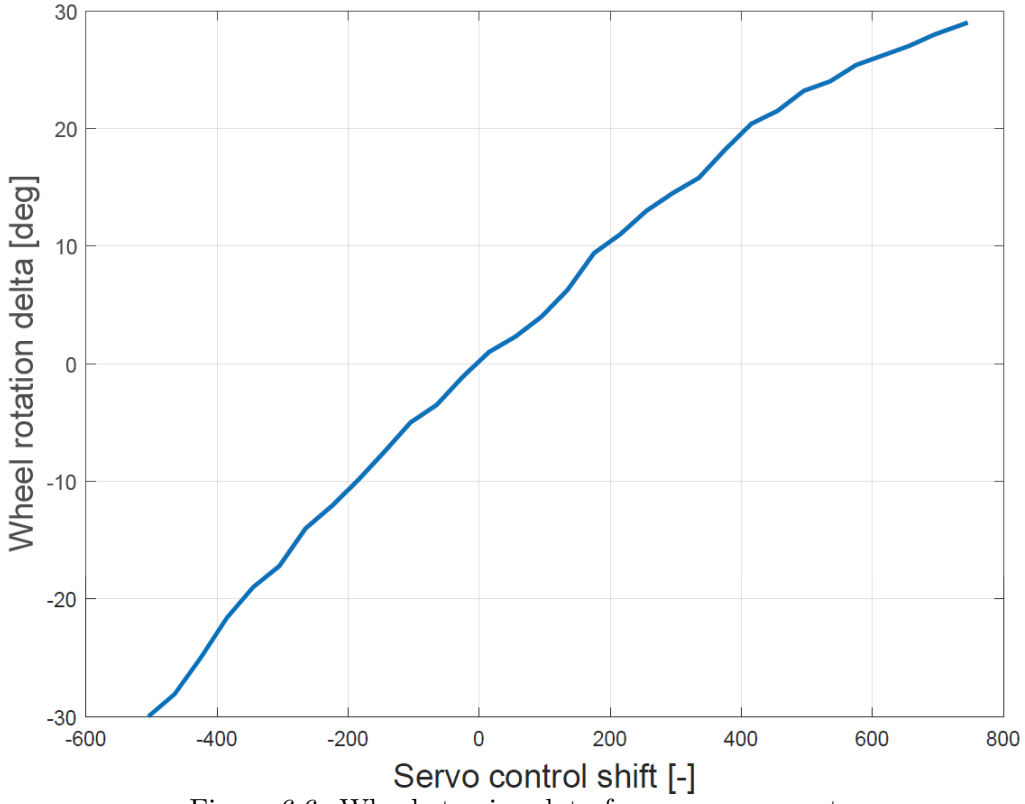


Figure 6.6: Wheel steering data from measurement

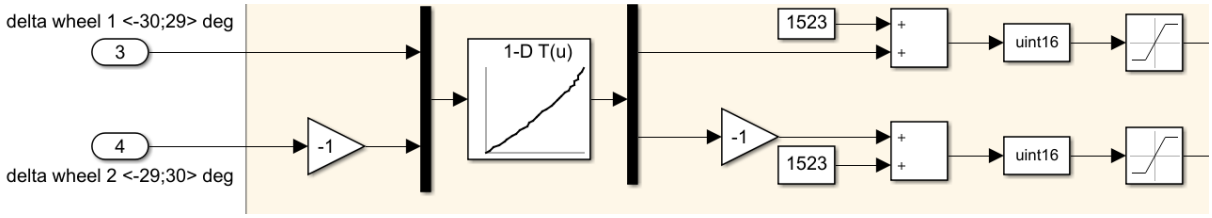


Figure 6.7: Wheel steering Look-up table in simulink

The servomotors are controlled by the duty cycle of the generated PWM signal with the period of $20\,ms$. In order to set the servomotor's shaft to the zero position (middle of the operating range) the PWM signal must have duty cycle of 7.5% ($1.5\,ms$). From there the servomotor's shaft has an operating range of $\pm 90°$ controlled by the shift from the zero position of the generated PWM signal which can have value of $\pm 3\,\%$ ($\pm 0.6\,ms$). So in order to control the servomotor's shaft's rotation an output compare module was used. The servomotor is then controlled by inputting the shift acquired from the 1D look-up table based on the required wheel's rotation in a summation block together with the value for the zero position computed using eq. 6.4. Equation 6.3 serves to calcute the needed input in order to achieve the wanted duty cycle where the OC_{max} is the maximum input

value of the output compare module (acquired from the *dsPIC33EP512MU810* and based on the set IPS), T_{ON} is the wanted amount of time of the duty cycle and the T is the period of the PWM. The implemented computation can be seen in fig. 6.7.

$$OC_{in} = OC_{max} * \frac{T_{ON}}{T} \tag{6.3}$$

$$OC_0 = 20311 \frac{1.5\,ms}{20\,ms} = 1523 \tag{6.4}$$

6.3 Ackerman geometry

The trigonometric functions used by the Ackerman geometry described in subsection 4.2.1 require too much computing power to implement directly. In order to implement the Ackerman geometry the kinematic model had been replaced using look-up tables.

Since only two ISL boards were available possibilities were limited to 2WD. After considering the possible options it has been decided that car4 would be setup as front axle 2WD and 2WS vehicle. This means that the ICR would always lie on the line defined by the rear wheels' axis of rotation. In the kinematic model described in subsection 4.2.1 this state could be achieved by setting the input parameter a which represents the position of ICR along the x axis to $a = 0$ or $a = l$ where l is the length of car4 (tab. 4.2). The only difference then would be the output indices and the orientation of the model. Further in the development the case of $a = 0$ was used (index 1 of the model represented the left driving wheel and index 3 of the model represented the right driving wheel).

The Ackerman geometry could then be described as a system with two inputs and four outputs. Two 2D look-up tables could be used but such implementation would require a lot of computing power. After thorough inspection of the equations describing the model (eqs. 4.18 to 4.36) considering the simplification of constant parameter a and that the rear wheels will hold their default position and are not driven several properties allowing a less computation heavy solution were discovered.

In the case of outputs δ_1 and δ_3 as expected when the input parameter a was to be set to $a = 0$ the two other outputs became 0. The required two angles δ_1 and δ_3 then became dependent on one input parameter R which is the distance of ICR along the y axis (eqs. 6.5 and 6.6).

$$\delta_1 = f(R) = \arctan \frac{0.5}{R - 0.2} \tag{6.5}$$

$$\delta_3 = f(R) = \arctan \frac{0.5}{R + 0.2} \tag{6.6}$$

This meant that the process of calculating δ_1 and δ_3 could be viewed as two functions each with one input and one output and could be implemented using two 1D look-up tables. Another feature that was used in order to further decreases the amount of required computation power was the similarity between the functions $\delta_1 = f(R)$ and $\delta_3 = f(R)$. Since the constants in the equations were of the same value and differ only in one sign only one look-up table was implemented for the purpose of computing both δ_1 and δ_3. The first angle was received by passing the required R and the other angle was then acquired using the equation 6.7. Using these features the computation of δ_1 and δ_3 was

implemented with a single 1D look-up table using the function $\delta_1 = f(R)$.

$$\delta_3 = \arctan\frac{0.5}{R+0.2} = -\arctan\frac{0.5}{-R-0.2} \tag{6.7}$$

Similar simplification was achieved for the outputs ω_1 - ω_4 (eqs. 4.28 - 4.31). Since only the front axle was driven the outputs ω_2 and ω_4 were ignored. The other two outputs required both remaining inputs R and v. Fortunately the whole calculation process could be simplified into equation 6.8 where $\alpha_{AGi} = f(R)$ was a function of the input parameter R only and $i \in \{1, 3\}$ and was called wheel velocity coefficient. By multiplying the α_{AGi} by the input velocity the required ω_i could be attained.

$$\omega_i = \alpha_{AGi}\, v \tag{6.8}$$

$$\alpha_{GAi} = \frac{1}{|\cos\beta|}\frac{R_i}{R_0}\frac{1}{r} \tag{6.9}$$

This wheel velocity coefficient was comprised of three transformations (eq. 6.9). The first transformation recalculated the input forward velocity of the vehicle based on the placement of the ICR ($\frac{1}{|\cos\beta|}$ where β is the angle between the wanted forward velocity and the actual direction of the velocity dictated by the placement of the ICR). The second transformation recalculated the velocity along the circle with the center defined by the ICR of the vehicle's center of mass to the needed wheel velocity to follow the circle centered in the ICR and maintain the wanted forward velocity ($\frac{R_i}{R_0}$ where R_i is the wheel to ICR distance and R_0 is the vehicle's center of mass to ICR distance). And the third transformation returned the needed angular velocity ω_i from the needed velocity of the wheel ($\frac{1}{r}$ where r is the radius of the vehicle's wheels).

The model presumes symmetrical vehicle (equal distance of wheels from the center of mass). This fact leads to a possible simplification described by equation 6.10. Using this simplification the computation of ω_1 and ω_3 was reduced to a single 1D look-up table using the function $\alpha_{AG1} = f(R)$.

$$\alpha_{AG3}(R) = \alpha_{AG1}(-R) \tag{6.10}$$

The implementation of the Ackerman geometry in simulink using the two look-up tables can be seen in fig. 6.8.

The two 1D look-up tables required the functions $\delta_1 = f(R)$ and $\alpha_{AG1} = f(R)$ which were inspected and divided into 7 intervals depending on their apparent Non-linearity and also based on car4's limitations:

- Interval 1 - Small Non-linearity, increasing $R \in (-120; -12]$
- Interval 2 - High Non-linearity, increasing $R \in (-12; -2.5]$
- Interval 3 - Small Non-linearity, increasing $R \in (-2.5; -1.06]$
- Interval 4 - Physically unattainable region $R \in (-1.06; 1.06]$
- Interval 5 - Small Non-linearity, decreasing $R \in (1.06; 2.5]$

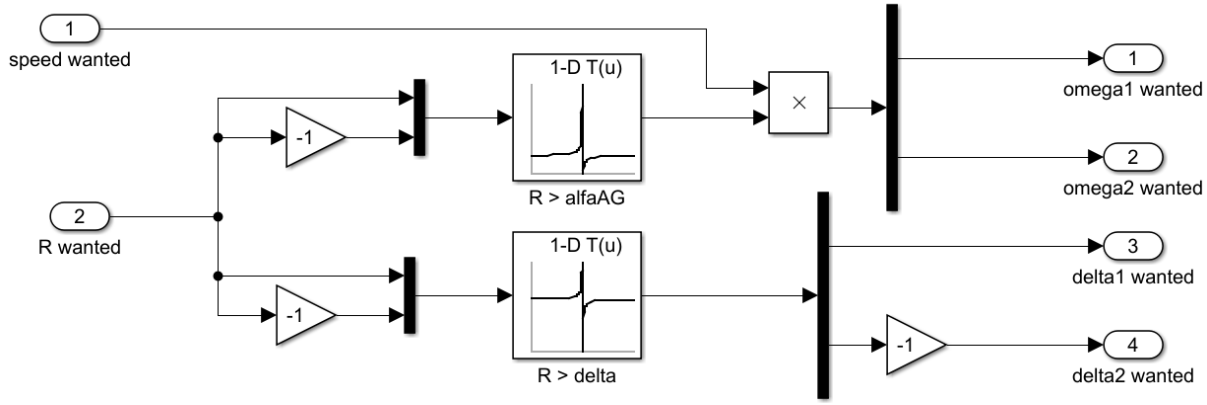


Figure 6.8: Ackerman geometry using look-up tables in simulink

- Interval 6 - High Non-linearity, decreasing $R \in (2.5; 12]$
- Interval 7 - Small Non-linearity, decreasing $R \in (12; 120]$

The interval of physically unattainable setups was stated by finding out the corresponding value of input parameter $R = 1.06\,m$ when the inner wheel reaches it's maximum rotation $\delta = 30°$.

Different amounts of linearly spaced points denser in intervals 2 and 6, thinner in intervals 1,3,5 and 6 and empty in interval 4 were created. These sets were then used as inputs to functions $\delta_1 = f(R)$ and $\alpha_{AG1} = f(R)$ forming the needed datasets needed for the creation of the look-up tables.

A simulink model for testing was created. This model used "From workspace" block to load input data from matlab and pass the data through the two 1D look-up tables setup using the created datasets and then the output data was send back to matlab using the block "To workspace."

Then the same input data used in the simulink model were passed through the created kinematic model described in subsection 4.2.1. The difference between these two outputs were then squared and the resulting error was logged. After several iterations of changing the number of linearly spaced components in the intervals based on the computed error two sets of data each for the functions $\delta_1 = f(R)$ and $\alpha_{AG1} = f(R)$ were chosen. These datasets have 56 points and their placement plotted with the functions $\delta_1 = f(R)$ and $\alpha_{AG1} = f(R)$ and the final error of the look-up tables can be seen in figures 6.9 and 6.10. The error of the look-up table computing the wheel rotaion δ has maximum value of less the 0.04 and the error of the look-up table computing the wheel velocity coefficient α_{AG} has maximum value of less then 0.001. These errors were considered small enough not to cause problems.

In order to be able to drive the car4 forward an additional point was inserted into the two look-up tables. When the input parameter $R = 0$ the output rotation angle $\delta = 0$ and the wheel velocity coefficient $\alpha_{AG} = \frac{1}{r}$ which solves the singularity of the algorithm when $R = inf$.

This implementation was tested by setting up fixed starting position and then uploading the control algorithm set to certain value of R. Since the ICR always lied on the axis of the rear axle perpendicular line from the rear wheels' starting position was setup and after the car4 moved some distance another perpendicular line was lead from the rear wheels until it crossed the perpendicular line from the starting position. The length of the

line was measured and $w/2 = 0.2\,m$ was added in order to receive the actual value of R. The results can be seen in tab. 6.2. Both tests yielded slightly bigger R than the required most probably caused by the inaccuracy of measured data used for steering control.

R_{wanted} $[m]$	R_{mes1} $[m]$	R_{mes2} $[m]$	R_{mes3} $[m]$	R_{mes4} $[m]$	R_{mes5} $[m]$
1.10	1.15	1.10	1.17	1.21	1.15
2.00	2.20	2.17	2.23	2.27	2.15

Table 6.2: Results of steering tests

The DC motors' shafts were connected to gearboxes with gearbox ratio 1:14. In order to make the speed control functional the required speed was multiplied by 14. Test were performed by preparing markers at measured positions ($0\,m$, $0.5\,m$, $1\,m$, $1.5\,m$ and $2\,m$) through which car4 has been driven at set speeds. These drives were recorded and then the times to reach each markers were measured. Using these measurements the average speeds between the markers were calculated. The results can be seen in tab. 6.3. Based on the test the speed control was deemed satisfactory.

v_{wanted} $[ms^{-1}]$	v_1 $[ms^{-1}]$	v_2 $[ms^{-1}]$	v_3 $[ms^{-1}]$	v_4 $[ms^{-1}]$
0.2	0.192	0.205	0.207	0.198
0.5	0.52	0.50	0.48	0.54

Table 6.3: Results of velocity tests

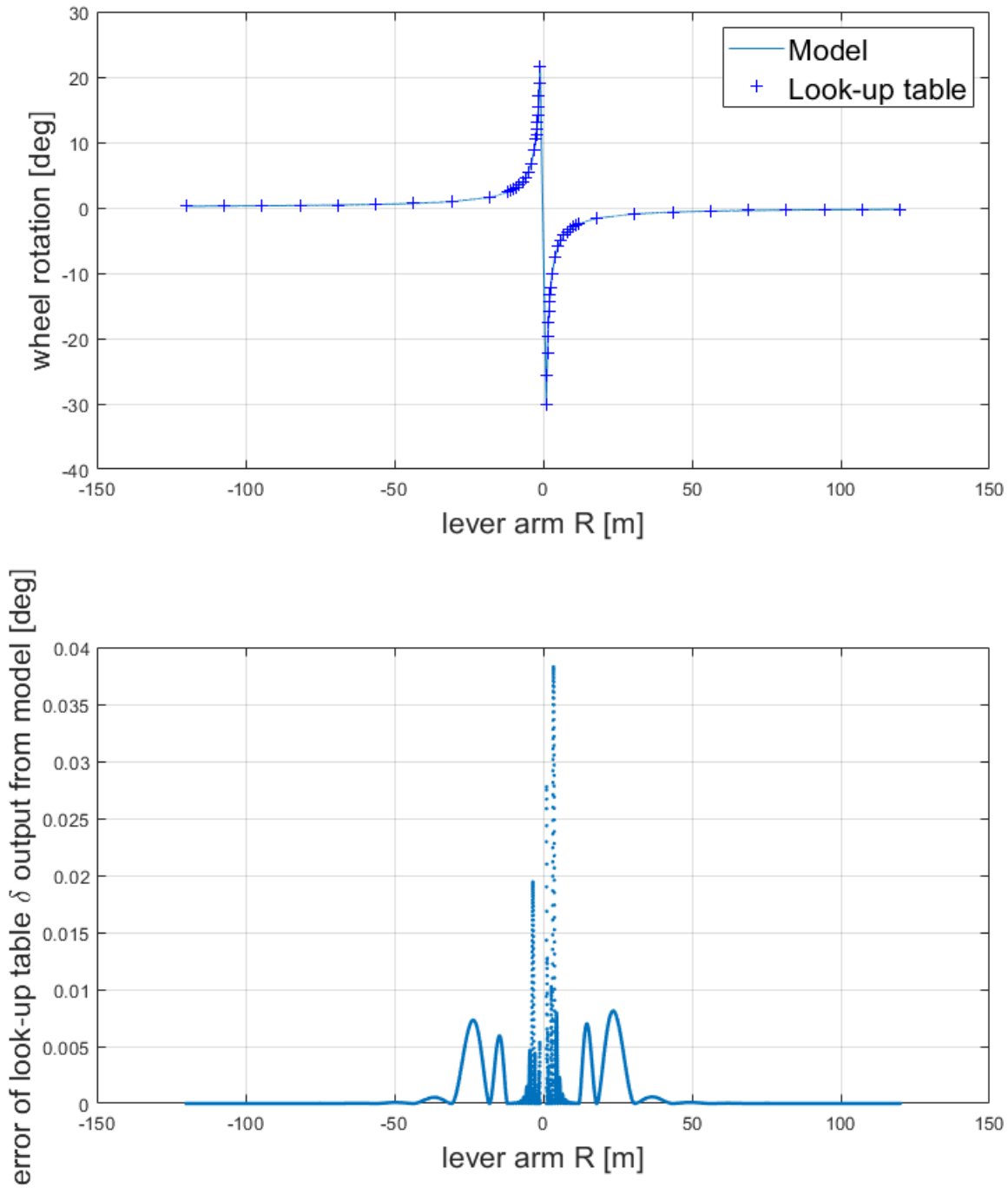


Figure 6.9: Validation of look-up table replacing ackerman geometry rotation computation

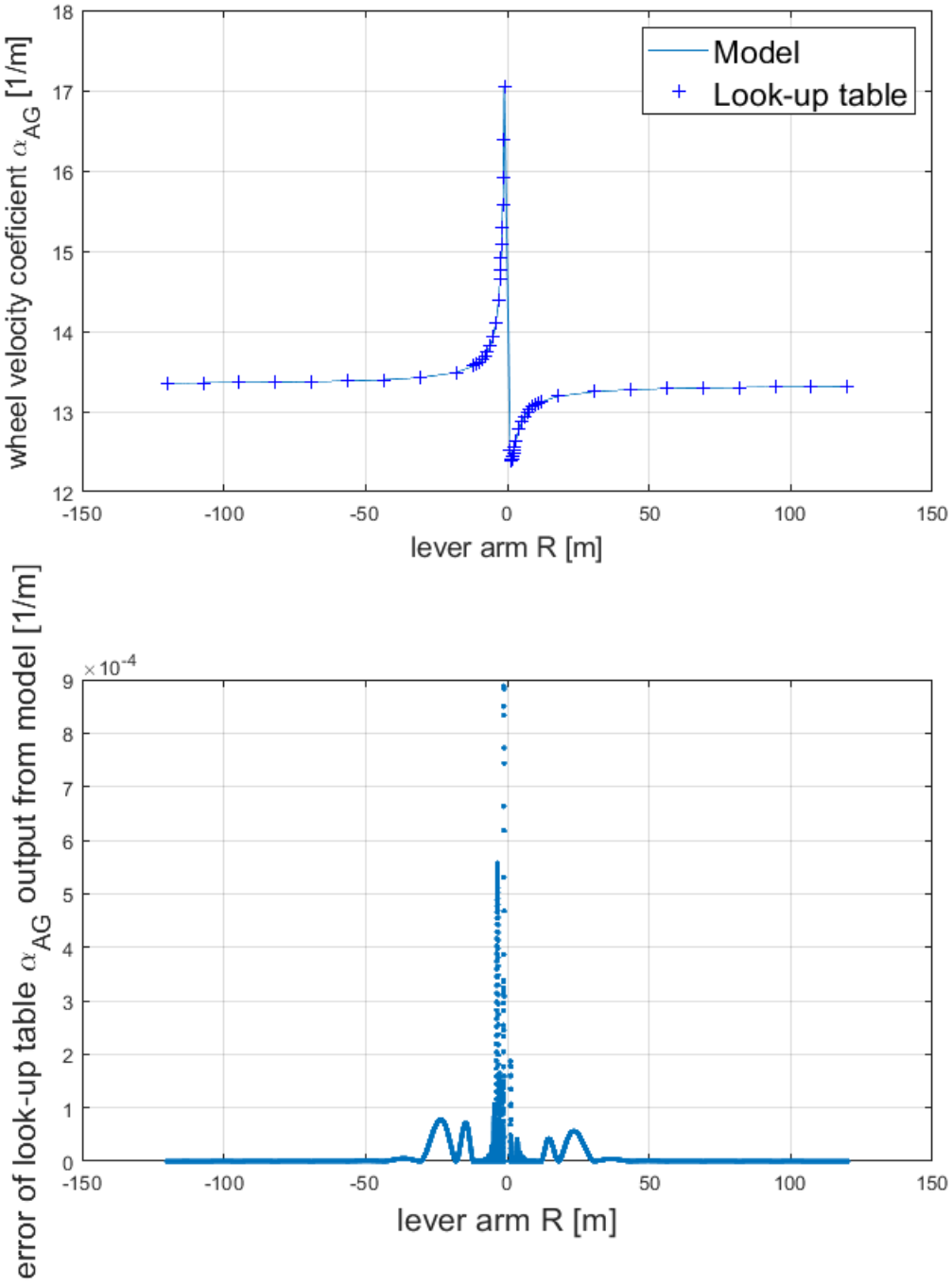


Figure 6.10: Validation of look-up table replacing ackerman geometry velocity computation

7 Conclusion

The first goal of this thesis was to revise the state of electronics on the robotic vehicle car4 with focus on wheels' speed of rotation measurement, power electronics and signal electronics. The second goal of this thesis was to develop a control algorithm using a kinematic model that ensures independent adjustment of the speed of rotation of wheels and also ensures no-slip steering.

In the first part of this work theoretical survey of kinematic models focusing on 2WS/2WD and 4WS/4WD vehicles was done. Two types of kinematic models holonomic and nonholonimic and the difference between them was described. Examples of both holonomic and nonholonomic models of wheeled vehicles were then presented.

The state of the present electronics was analyzed. The used MCU was the original from 2010 and was capable of communicating with the other units on car4 which were: RC module, Magnetic encoders, ISL boards and the servo interface units. Although it could control only 2 DC motors and 2 servomotors. Two ISL boards with current sensors from the original design were present on car4 each capable of controlling 1 DC motor. Two servo interface units also from the original design were present on the vehicle each capable of controlling 2 servomotors. Finally four magnetic encoders using I2C communication each for one motor were present on the car4. For the purpose of adjusting the supply voltage from the batteries two power units were present on the vehicle. One dedicated for servomotors and one for the control electronics. The schematic of the car4's electronics at the beginning of this thesis can be seen in fig. 3.5. The vision for future development of car4 was proposed and a list of required changes was formulated.

Two developed kinematic models and the tests performed on these models were then described. The first model represented the car4's steering mechanism. This model was created because the old model didn't correspond to the real mechanism and accurate control of the steering was needed. The second model was that of a 4WD/4WS vehicle using Ackerman geometry.

The design process of new electronics started by formulating the requirements and creating a design schematic of the car4's electronics (fig. 5.1). Lists of requirements for the new MCU, 4x new H-bridge units and 1x new servo interface unit were formulated. The design processes of the MCU and the servo interface unit were then described. The H-bridge units were not manufactured in the end due to their complexity and lack of time. This meant that the electronics implemented on the vehicle differed from the formulated design schematic. The difference can be seen by comparing the design schematic with the schematic of implemented electronics in fig. 5.7.

In the final chapter the implementation of the new control algorithm and the performed tests were described. This section was divided into tree separate parts focusing on Wheel rotation speed control, wheel steering control and Ackerman geometry. In the case of wheel rotation speed a PI controller was used because it showed best performance even when the DC motor experienced voltage drops when the draw current suddenly increased.

These drops were caused most likely by battery pack's wear. During the testing of new wheel steering model described in section 4.1 it has been discovered that both the old and the new models are non-optimal because the mechanism cannot be simplified as 2D mechanism. In order to control the steering measurement of input signal to wheel rotation was performed. The acquired data was used to create a 1D look-up table for wheel steering control. Finally analysis of the 4WS kinematic model using Ackerman geometry which is a algorithm with 3 inputs and 8 outputs was done. Using different features and simplifications the model was reduced to two 1D look-up tables which greatly saved computing power of the on board MCU.

7.1 Suggestions for future development

In the future the development of car4 might focus on:

- Adding a RS232-TTL converter onto the MCU for the communication with the RC module.
- New H-bridge units to control the DC motors are needed. The design of new H-bridge units with notes are part of this thesis's attachments.
- Revision of car4's battery pack, possibly adding a port for balanced charging.
- Creating new model of steering mechanism, possibly using simscape or the created solidworks model.

List of Figures

List of Abbreviations

FME	Faculty of mechanical engineering
BUT	Brno university of technology
SLAM	Simultaneous localization and mapping
DOF	Degree of freedom
2WS	2 wheel steering
4WS	4 wheel steering
2WD	2 wheel drive
4WD	4 wheel drive
MCU	Main control unit
PCB	Printed circuit board
ICR	Instant center of rotation
ABS	Anti-lock braking system
ASR	Anti slip regulation
PID	Proportional–integral–derivative controller
PWM	Pulse width modulation
DIR	Direction
DIS	Disable
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
SSI	Synchronous Serial Interface
UART	Universal asynchronous receiver-transmitter
GPIO	General purpose input output
RAM	Random Access Memory
HW	Hardware

SDA Serial Data

SCL Serial Clock

RC Remote control

IC Integrated circuit

IPS Instructions per second

Bibliography

[1] DOBOSSY, Barnabás, 2019. *Segway driver parameter estimation and its use for optimizing the control algorithm* [online]. Brno [cit. 2021-5-15]. Available from: `https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=193560`. Master's thesis. Brno university of technology, Faculty of mechanical engineering, Institute of solid Mechanics, Mechatronics and Biomechanics. Supervisor: Ing. Martin Brablc.

[2] BRABLC, Martin, 2014. *Traction control of the experimental vehicle with four steering and driven wheels* [online]. Brno [cit. 2021-5-15]. Available from: `https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=87781`

[3] JASANSKÝ, Michal, 2010. *Design of dynamic models for traction control of experimental vehicle* [online]. Brno [cit. 2021-5-15]. Available from: `https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=28793`. Master's thesis. Brno university of technology, Faculty of mechanical engineering, Institute of solid Mechanics, Mechatronics and Biomechanics. Supervisor: Ing. Robert Grepl, Ph.D.

[4] ŠŮSTEK, Jan, 2020. *Control unit design for robotic vehicle car4* [online]. Brno [cit. 2021-5-15]. Available from: `https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=213718`. Bachelor's thesis. Brno university of technology, Faculty of mechanical engineering, Institute of solid Mechanics, Mechatronics and Biomechanics. Supervisor: Barnabás Dobossy.

[5] VADLEJCH, Filip, 2010. *Design of experimental vehicle undercarriage with four wheel steering* [online]. Brno [cit. 2021-5-15]. Available from: `https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=26364`. Bachelor's thesis. Brno university of technology, Faculty of mechanical engineering, Institute of solid Mechanics, Mechatronics and Biomechanics. Supervisor: Ing. Robert Grepl, Ph.D.

[6] VEJLUPEK, Josef, 2010. *Development of electronics for traction control of experimental vehicle* [online]. Brno [cit. 2021-5-15]. Available from: `https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=29115`. Master's thesis. Brno university of technology, Faculty of mechanical engineering, Institute of solid Mechanics, Mechatronics and Biomechanics. Supervisor: Ing. Robert Grepl, Ph.D.

[7] CHARAKOV Dimitar, 2006. Kinematics Model of Nonholonomic Wheeled Mobile Robots for Mobile Manipulation Tasks. In: 5th BALTIC - BULGARIAN CONFERENCE ON BIONICS AND PROSTHETICS [online]. Varna : 2006, s. 59-61 [cit. 2021-5-15]. Available from: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.557.1868&rep=rep1&type=pdf`

[8] MIN WAN CHOI, Jun Seok PARK, BONG SOO LEE a MAN HYUNG LEE, 2008. The performance of independent wheels steering vehicle(4WS) applied Ackerman geometry. In: 2008 International Conference on Control, Automation and Systems [online]. Seoul: IEEE, 2008, s. 197-202 [cit. 2021-5-15]. ISBN 978-89-950038-9-3. Available from: doi:10.1109/ICCAS.2008.4694549

[9] FLORIAN, Zdeněk, Emanuel ONDRÁČEK a Karel PŘIKRYL, 2007. *Mechanika těles: statika* [online]. Vyd. 7., V Akademickém nakladatelství CERM 2. Brno: Akademické nakladatelství CERM. ISBN 978-80-214-3440-0.

[10] GREPL, Robert, 2007. *Kinematika a dynamika mechatronických systémů*. Brno: Akademické nakladatelství CERM. ISBN 978-80-214-3530-8.

[11] GREPL, Robert, 2008. *Mechatronika - vybrané problémy*. Brno: Tribun. ISBN 978-80-214-3804-0.

[12] PŘIKRYL, Karel, 2008. *Kinematika*. Vyd. 5., V Akademickém nakladatelství CERM 3. vyd. Brno: Akademické nakladatelství CERM. ISBN 978-80-214-3679-4.

[13] ams AG [online catalog list]. AS5048A/AS5048B. ©2018 [cit. 2021-5-15]. Available from: `https://ams.com/documents/20143/36005/AS5048_DS000298_4-00.pdf/910aef1f-6cd3-cbda-9d09-41f152104832`

[14] Future Technology Devices International Ltd. [online catalog list]. FT231X. ©2013 [cit. 2021-5-15]. Available from: `https://ftdichip.com/wp-content/uploads/2020/08/DS_FT231X.pdf`

[15] Hitec RCD [online catalog list]. General Servo Information. ©2012 [cit. 2021-5-15]. Available from: `http://www.hitecrcd.com/files/Servomanual.pdf`

[16] Microchip Technology Inc. [online catalog list]. dsPIC33EPXXX(GP/MC/MU)806/810/814 and PIC24EPXXX(GP/GU)810/814. ©2012 [cit. 2021-5-15]. Available from: `https://ww1.microchip.com/downloads/en/DeviceDoc/70616g.pdf`

[17] Microchip Technology Inc. [online catalog list]. dsPIC33FJ32MC302/304, dsPIC33FJ64MCX02/X04 and dsPIC33FJ128MCX02/X04. ©2012 [cit. 2021-5-15]. Available from: `https://ww1.microchip.com/downloads/en/DeviceDoc/70291G.pdf`

[18] Silicon Laboratories, Inc. [online catalog list]. Si864x Data Sheet. ©2019 [cit. 2021-5-15]. Available from: `https://www.silabs.com/documents/public/data-sheets/si864x-datasheet.pdf`

[19] Texas Instruments [online catalog list]. LMS8117A. ©2016 [cit. 2021-5-15]. Available from: `https://www.ti.com/lit/ds/symlink/lms8117a.pdf?ts=1621085854525&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLMS8117A`

[20] Transmotec [online catalog list]. PD4266. ©2020 [cit. 2021-5-15]. Available from: `https://www.transmotec.com/Download/Catalog/Transmotec-EN-PD-0.6W-80W-2020.pdf`

[21] Převodník TTL na RS232, modul s MAX3232. Martin Lhotský - prodej elektrosoučástek [online]. [cit. 2021-5-15]. Available from: https://www.lhotsky-elektro.cz/Prevodnik-TTL-na-RS232-modul-s-MAX3232-d46815.htm?tab=download

[22] Seawise Giant, 2001-. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation [cit. 2021-5-19]. Dostupné z: https://en.wikipedia.org/wiki/Seawise_Giant